R^2SS 기반의 정보검색 시스템

Information Retrieval System for R^2SS

홍석주, 박영배
명지대학교 컴퓨터공학과

Seok-Joo Hong(sjhong@nate.com), Young-Bae Park(parkyb@mju.ac.kr)

요약

본 논문은 R^2SS((Reverse Really Simple Syndication) 기반의 지능형 검색엔진의 설계 및 구현에 관한 것으로, 기존의 방식과 같이 사용자가 RSS 주소를 입력하여 제한된 RSS 정보를 받아보는 방식이 아니라, 사용자는 단순히 자신이 원하는 정보를 입력만 하면, 자동화된 RSS 주소수집서버가 수집한 수많은 RSS 주소들로부터 실시간으로 수집하는 RSS 규격 문서들 중 사용자가 원하는 규격 문서에 대한 RSS 정보만을 제공해줌으로써, 수많은 정보를 찾아 그 중 원하는 정보만 추려서 제공해주는 R^2SS 구독(Reverse RSS Subscribe) 방식을 설계하는데 있다. 제안된 R^2SS 기반 지능형 검색엔진을 통하여 양질의 정보를 찾아서 헤매는 시간을 획기적으로 줄일 수 있고 개인 비서를 두게 되는 효과를 얻을 수 있다.

■ 중심어 : | 알에스에스 | 역 알에스에스 | 역 알에스에스 구독기 | 크롤링 |

Abstract

This study matters the design and implementation of an intelligent information search engine that is based on the R^2SS(Reverse Really Simple Syndication). Apart from to the previous method, where the user inputs the RSS address that one intends and obtains limited RSS information, the user just types in the information that one appoints to acquire the RSS information of standard documents that the user is interested among several RSS addresses by a Reverse RSS(Really Simple Syndication) method, which is drawn by the automated RSS address collection server in realtime. Through the proposed R^2SS(Really Reverse Simple Syndication) based intelligent information search engine, time can be significantly saved along with obtaining information with good quality, furthermore, it has the effects of having a personal secretary.

■ keyword : | RSS(Really Simple Syndication) | Reverse RSS | RSS Reader | Crawling |

I. 서론

기존의 RSS 리더기(Reader)는 사용자가 RSS 주소를 직접 입력하면 해당 RSS 주소를 주기적으로 방문하여 새로운 정보가 올라온 경우 사용자에게 알려주었다. 이러한 RSS 리더기를 사용한 방식은 사용자가 해당 RSS 주소를 직접 알아야 정보를 받아 볼 수 있다는 사용 편리성에 있어서 치명적인 단점과 각 사용자들이 개별적

접수번호 : #091105-002
접수일자 : 2009년 11월 05일
심사완료일 : 2009년 11월 20일
교신저자 : 홍석주, e-mail : sjhong@nate.com

으로 알고 있는 RSS 주소의 수가 많지 않다는 단점이 존재하고, 이러한 방식으로 인해서 커다란 활용 여지가 있음에도 불구하고 RSS 리더기는 사용자층을 많이 확보하지 못하였다[14][16][17].

또한, 기존의 RSS 리더기가 가지는 두 번째 문제점을 해결하기 위한 것 중 하나가 메타 블로그(Meta Blog)로서 이는 여러 사람이 수동으로 입력한 RSS 주소를 공유해서 다양한 RSS 주소로부터 콘텐츠를 가져와서 사용자에게 보여 주거나 검색할 수 있도록 한다. 이렇게 RSS 주소가 좀더 많아지기는 하였지만 여전히 사용자가 수동으로 입력한 극소수의 RSS 주소에 의존하고 있다. 또한, 사용자는 다양한 사람들이 입력한 RSS 주소로부터 정보를 받아 보면서 그 중에 자신이 원하는 정보를 선택해야 하는 수고를 해야 한다[1][2].

본 논문은 이러한 문제점을 해결하기 위하여 제안한 것으로서, 본 논문의 목적은 RSS 규격 문서를 기존에 사용자가 직접 RSS 주소를 미리 알고 있으면서 입력도 해야 하는 불편함을 역 RSS 구독 방식을 사용하여 편이성과 유용성을 증가시킨 역 RSS(R^2SS) 기반 지능형 정보 검색시스템을 제공하는데 있다.

본 논문의 또 다른 목적은 RSS 리더기 부분에서는 기존의 방식과 같이 사용자가 RSS 주소를 입력하여 제한된 정보를 받아 보는 방식이 아니라 사용자는 단순히 자신이 원하는 정보를 입력만 하면, 자동화된 RSS 주소수집서버가 수집한 수많은 RSS 주소들로부터 실시간으로 수집하는 RSS 정보들 중에서 사용자가 원하는 정보에 대한 역 RSS 문서 정보를 제공하여 RSS의 사용 용이성 한계와 제공되는 정보 범위의 한계를 극복할 수 있도록 한 역 RSS 기반 지능형 정보 검색 시스템을 제공하는데 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 이론적 고찰을 위하여 기존 RSS 특징과 문제점, 웹 크롤러, 웹 문서 순위화를 살펴본다. 3장에서는 기존 RSS를 이용한 정보검색시스템의 문제점을 해결하기 위한 R^2SS 기반의 지능형 정보검색시스템의 설계와 기술요소를 설명한다. 4장에서는 본 제안 시스템의 성능 분석을 위하여 메타 블로그 사이트와의 성능 비교, R^2SS 출처 순위화의 성능 평가, 그리고 R^2SS 색인어에 따른 성능 평가를 제시한다. 마지막으로 5장에서는 결론을 맺는다.

II. 관련 연구

1. RSS(Really Simple Syndication)

Really Simple Syndication, 혹은 Rich Site Summary의 줄임말이며, XML(eXtensible Markup Language) 혹은 RDF(Resource Description Framework) 기반의 콘텐츠 배급 프로토콜이다[12][13]. 이는 웹정보 제공자 측에서 새로운 정보의 갱신 여부를 알려주는 용도로 쓰인다. 현재 RSS는 [표 1]과 같이 7가지 버전이 존재한다[12][15].

표 1. RSS의 버전별 특징

버전	오너	설명	진행
0.90	넷스케이프	초기 버전	1.0에 의해 중단
0.91	유저랜드	0.90 간략화	2.0에 의해 중단되었지만, 많이 쓰임
0.92, 0.93, 0.94	유저랜드	0.91확장	2.0에 의해 중단
1.0	RSS개발 그룹	RDF기반	안정화코어, 모듈개발 진행 중
2.0	유저랜드	0.91확장	안정화코어, 모듈개발 진행 중

RSS 기술을 이용하면 [그림 1]과 같이 신규 자료를 찾기 위해 해당 사이트를 반복적으로 접근할 필요가 없다. 사용자는 RSS Aggregator (Reader)라는 프로그램에 해당 RSS 채널의 주소만 기억해두면, 나중에는 RSS를 통하여 자동으로 신규 자료로 판단된 정보가 사용자에게 전달되므로 웹 정보를 습득하는데 있어 사용자의 수고가 크게 줄어든다.

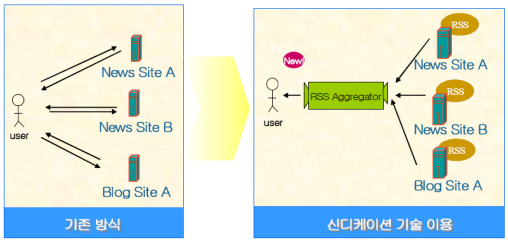


그림 1. RSS의 기술적 변화

RSS는 1인 미디어 플랫폼인 블로그의 확산과 함께 늘어나고 있는 추세이다. 2007년 PEW ONTERNET 조사에 따르면 미국의 1억 2천만 성인 인터넷 사용자 중 7%가 자신의 블로그를 만든 경험이 있고(800만 명), 27%가 블로그를 주기적으로 구독하고 있으며, 5%가 RSS Aggregator를 사용하고 있다[1][2].

2. 웹 크롤러

인터넷 사용자들은 원하는 정보를 찾기 위해 검색 사이트를 이용한다. 이러한 검색 사이트는 내부 DB를 가지고 있으며, 이러한 DB 구축을 위해 웹 크롤러가 동작한다. 웹 크롤러란 URL을 따라서 이동하면서 웹상의 정보를 수집하는 일종의 소프트웨어이다. 하지만, 웹 상에 존재하는 정보에 접근하기 위한 URL 목록 저장소는 따로 존재하지 않는다. 따라서 크롤러가 웹 페이지에 존재하는 정보를 가져오기 위해 URL을 수집하는 유일한 방법은 웹 페이지에 존재하는 하이퍼링크를 이용하여 아직 수집되지 않은 다른 URL에 접근하는 방법을 취한다.

크롤러는 주어진 초기 URL집합으로 시작해서 점진적으로 새로운 URL을 얻기 위해 웹 페이지에 존재하는 링크를 수집하고, 차례로 무한 반복하여 정보를 수집한다. 이러한 방식의 이동은 웹 상에 자신의 가리키는 링크가 없는 문서에는 크롤러가 접근하지 못하는 한계점을 가진다. 이러한 링크를 따라 탐색할 수 없는 영역을 히든 영역이라 부르며, 이에 접근하여 정보를 수집하기 위한 웹 크롤러 구현에 관한 연구가 진행 중에 있다[14][15].

웹 크롤러의 구현에서 고려되는 사항은 DNS서버에 대한 병목 현상 최소화, 웹 사이트에 대한 부하 감소, URL의 중복 처리, 문서 내용의 중복처리, 문서 탐색 방법 등이 있다[14][16].

3. 웹 문서의 순위화

대부분의 탐색 엔진들은 순위화를 위하여 불리안과 벡터 모델의 변형을 이용한다. Yuwono와 Lee[18]는 고전적인 TF-IDF 방식 외에 세 가지 순위화 알고리즘을 제안했는데, 이들은 Boolean Spread, Vector Spread, Most-Cited라 불린다.

처음 두 개는 응답 내의 한 페이지가 가리키는 페이지들 혹은, 응답 내의 한 페이지를 가지키는 페이지들을 포함하기 위해 확장된 불리안과 벡터 모델의 정규 순위화 알고리즘들이다. 세 번째에 있는 Most-Cited는 응답 내의 페이지들에 대한 링크를 갖고 있는 페이지들에 포함된 용어들에 기반 한다. 2400개의 웹 페이지들로 구성된 컬렉션에 대하여 56개의 질의들을 고려하여 이들 기법들을 비교하면, 벡터 모델이 평균 75%의 정확률을 가진 더 좋은 재현율-정확률 곡선을 나타낸다.

HITS(Hypertext Induced Topic Search)에서는 하이퍼링크 정보를 이용하여 순위화 한다[19]. 한 페이지를 가리키는 하이퍼링크의 수는 평판도(popularity)와 질(quality)의 수준을 제공한다. 또한 페이지들에 공통적인 많은 링크들 혹은 동일 페이지에 의해 참조된 페이지들은 종종 그 페이지들 사이의 관계를 나타낸다. HITS에서는 응답에 있는 페이지들을 가리키거나 그 페이지들이 가리키는 페이지들의 집합 S를 고려한다. S에서 자신을 가리키는(들어오는) 많은 링크들을 갖고 있는 페이지들은 권위자(Authorities)라 하고, 나가는 링크들을 많이 갖고 있는 페이지들은 허브(Hub)라 불린다.

웹 문서들의 상대적인 중요성을 측정하기 위해 제안된 것이 페이지랭크(PageRank)이다. 권위자와 허브 값은 검색 시 매번 계산하는 반면 페이지랭크는 색인 작업 시 미리 계산하여 빠른 검색이 가능하다.

따라서 페이지랭크는 상업적인 정보 검색기에 더 많이 사용된다. 페이지랭크는 하이퍼링크를 통해 연결되어 있는 웹 문서들을 그래프 구조로 생각하고 웹 문서들의 순위를 계산한 것이다. 많은 문서들이 가리키고 있는 문서가 더 중요하다고 생각하고 그것에 대한 값을 수치로 표현한다.

4. 색인어 추출

텍스트 전문 표현 방법을 취하는 경우 텍스트에 나타난 모든 단어가 색인어로 사용된다. 다른 방법으로는 모든 단어를 색인어로 사용하지 않고 좀 더 추상적인 관점을 취하는 방법이 있는데, 이는 색인에 사용할 용

어 집합을 선정해야 함을 의미한다. 서지학 분야에서는 이러한 색인어 선정이 전문가에 의해 보통 이루어지는데, 다른 대안은 색인어 후보를 자동으로 선정하는 것이다[16].

자동 색인어 선정의 경우, 여러 가지 다른 접근 방법을 사용할 수 있는데, 이중 명사 집단 식별 방법을 예로 든다[16]. 자연언어 텍스트 문장은 일반적으로 명사, 대명사, 관사, 동사, 형용사, 부사, 접속사 등으로 구성되어 있다. 각 문법 범주에 속한 단어들이 나름대로 특정 목적에 사용되는 반면, 대부분의 의미는 명사에 의해 전달된다고 볼 수 있다. 따라서 색인어 선정에 있어서는 텍스트에 나타난 명사를 사용하는 것이 직관적으로 보아 타당한 전략이며, 이 방법에서는 명사를 제외한 나머지 단어를 체계적으로 제거하면 된다.

Ⅲ. R²SS 기반의 정보검색 시스템

1. 시스템 개요

본 논문에서 제안하는 R²SS 기반의 웹 크롤링 정보검색 시스템은 인터넷 상의 방대한 웹 콘텐츠를 기존의 방식과 같이 사용자가 RSS 주소를 입력하여 제한된 정보를 받아 보는 방식이 아니라, 사용자는 단순히 자신이 원하는 정보를 입력만 하면 자동화된 R²SS 주소 수집 처리가 수집한 수많은 RSS 주소들로부터 사용자가 원하는 정보에 대한 R²SS 문서 정보를 제공하여 RSS의 사용 용이성 한계와 제공되는 정보 범위의 한계를 극복할 수 있도록 한 R²SS 기반의 정보 검색 시스템이다.

R²SS 기반 웹 크롤링 시스템은 크게 R²SS 주소 수집 부분과 데이터 크롤링 부분으로 나누어진다. 제안한 R²SS 기반 웹 크롤링 시스템은 [그림 2]와 같으며 인터넷 상에서 RSS 주소들을 수집하여 각 사용자 정보들 및 각 사용자 요청 정보별 형태소들을 데이터베이스화하여 저장 및 관리한다. 이렇게 저장된 각 R²SS 파일에 대한 웹 문서 데이터의 형태소를 분석하여 각 사용자 요청별 형태소들과 비교한 후, 각 사용자 요청정보와 동일한 범주에 포함되는지의 여부를 판단하여 대용량 처리가 가능하도록 원격 객체 호출(RMI) 기능을 부여한다. 또한, 해당 사용자 요청 정보에 대한 R²SS 문서 정보를 해당 사용자 단말의 화면에 디스플레이 해주는 웹 서버도 포함하고 있다.

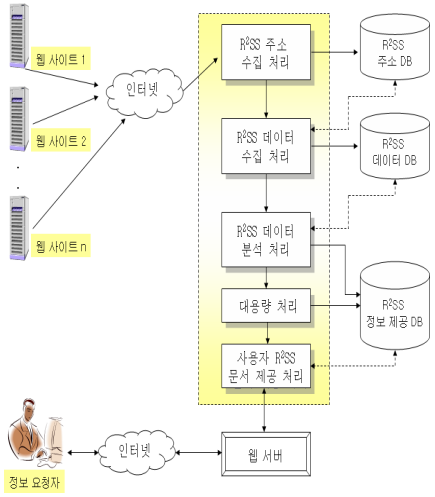


그림 2. R²SS 기반의 정보검색 시스템 플랫폼

2. R²SS 주소 수집 처리

R²SS 주소 수집 처리는 크게 블로그 주소 자동 수집 모듈과 R²SS 유효성 체크 모듈, 주소의 동적 디렉토리 구성 및 R²SS 주소 데이터베이스 엔진의 처리와 같은 단계를 거친다.

2.1 블로그 주소 자동 수집

R²SS 블로그 주소 자동 수집 처리는 블로그에 존재하는 RSS 출처 주소를 찾아서 수집해 온다. 블로그 주소 자동 수집 처리를 위해 고려된 사항은 다음과 같다. 첫째 블로그는 하나의 정형화된 구조를 띄지 않으며, 내용을 기반으로 구분되는 웹 페이지가 아니다. 따라서 웹을 대상으로 블로그 여부를 판단하는 것은 불가능에 가깝다. 둘째 RSS 출처 주소 링크는 대개 블로그 메인 페이지에 존재한다. 따라서 블로그에서 RSS 출처 주소를 찾기 위한 링크 탐색 범위는 블로그의 메인 페이지에 존재하는 링크 개수로 제안되어야 한다.

언급된 두 가지 고려사항을 해결하고 RSS 출처 수집 효과를 늘이기 위하여 본 논문에서 제시된 크롤러는 블

로그 서비스를 제공하는 포털 사이트의 랜덤 블로그 접근 모듈을 사용한다. 해당 모듈의 호출을 통해 크롤러는 지속적으로 블로그 주소를 획득하며, 블로그 메인 페이지에 존재하는 링크의 수로 탐색 큐의 사이즈를 제한하여 Breadth-First 방식으로 링크들을 조사한다.

단, 블로그 메인 페이지가 프레임 형태로 되어 있을 경우에는 추가 작업이 필요하다. 프레임에는 실제 링크 정보가 포함되어 있지 않으므로, 실제로 링크가 존재하는 웹 페이지를 찾기 위해선 프레임을 구성하고 있는 페이지로의 접근이 필요하다.

본 논문에서 제안한 R²SS 블로그 주소 자동 수집 처리는 메타 블로그와 같이 사람이 수동으로 블로그 주소를 등록하는 방식이 아니라, 로봇이 알아서 블로그 주소를 수집하는 알고리즘이다. 알고리즘의 작동원리는 댓글이나 트랙백 등 링크를 분석하여 새로 생성된 블로그를 자동으로 찾는 방식으로 작동한다.

2.2 R²SS 유효성 체크

R²SS 유효성 체크를 하기 위해서는 R²SS 파싱 모듈이 필수이다. 유효성 체크는 [그림 3]과 같이 유효한 블로그 주소를 체크한다. 이때 크롤러는 각 블로그 사이트 메인 페이지에 존재하는 태그들을 조사하기 위하여 "정규 표현식(Regular Expression)"을 이용한다[13].

```
// <a href=...> 태그에서 URL 매칭
_LINK_RE = re.compile(r'<\s*(a|img).+href|s*=\s*"?(.+?)"?(\s|>)',
                      re.MULTILINE | re.IGNORECASE)

# <frame src="bla"> 태그 매칭
_FRAME_RE =
re.compile(r'<\s*(frame).+src\s*=\s*"?(.+?)"?(\s|>)',
                      re.MULTILINE | re.IGNORECASE)

# HTML 문서의 타이틀을 조사
_TITLE_RE = re.compile(r'<\s*title.*?>(.+)</\s*title\s*>',
                      re.MULTILINE | re.IGNORECASE)

# GDS로 등록하기 위해 GUID를 plug-in한다.
_GUID = '{5e1788fe-a6e6-429f-816c-80cb969028d3}'
```

그림 3. 정규 표현식의 유효한 블로그 체크 분석기

해당 블로그가 프레임으로 되어 있을 경우, 각 프레임에 해당하는 주소를 획득하여 실제 블로그 메인 페이지 주소를 재설정한다. 블로그의 메인 페이지에 존재하는 링크들을 조사하기 위해 <a href> 형식의 태그에서 링크 주소를 획득한다. 크롤러는 획득된 링크 주소들에 접근하여 RSS 출처가 가져야할 태그 여부를 확인 후 RSS 출처라 판단되면, 기존에 이미 탐색된 출처인지의 비교를 통해 신규 출처일 경우 해당 RSS 출처 주소를 저장한다.

2.3 R²SS 주소의 동적 디렉토리 구성

습득된 R²SS 출처는 RSS를 통해 동적으로 디렉토리를 구성한다. 블로그가 어떤 디렉토리에 속하는지 여부는 사용자가 특정 질의 키워드를 모를 경우, 필요한 블로그를 찾는데 도움을 줄 수 있다. 현재의 블로그 서비스를 제공하는 포털 사이트나 여러 포털의 블로그 들을 한 데 모아서 제공하는 메타 블로그 사이트의 경우, 블로그별 디렉토리가 존재하지 않거나, 존재하는 경우에도 블로그 사용자가 가입 시 설정한 디렉토리로 고정되어 존재한다.

블로그의 특징 중 하나는 1인 미디어이므로, 사용자의 관심분야는 항상 바뀔 가능성을 내포하고 있다. 따라서 고정된 디렉토리는 해당 블로그의 특성을 적절히 나타내지 못할 가능성이 있다. 실제로 메타 블로그 서비스를 제공하며, 블로그의 디렉토리가 존재하는 Xpider 사이트[6]의 경우를 볼 때, 각 디렉토리별 상위 랭크된 블로그가 해당 디렉토리와는 맞지 않는 경우가 다수 존재한다.

이 같은 점을 보완하고자 본 논문에서는 [그림 4]와 같이 R²SS 문서에 포함된 아이템이 어떤 디렉토리에 속하는지 유사도 계산과 판단 방법을 통해 각각의 RSS 출처는 디렉토리별 점수를 가지게 되며, 이를 이용하여 디렉토리를 동적으로 구성하는 방법을 취한다.

```
public void AddChannel(ChannelData cData)
{
    if (!m_ChannelDAO.ExistXmlUrl(cData.XmlUrl))
    {

        // 아이템과 디렉토리 색인어가 유사도 계산
        iData.ItemDirIndex = Compute(item, directory)

        // 아이템과 디렉토리 선정
        cData.GroupIndex = xxxx(선정된 디렉토리의 인덱스)

        m_ChannelDAO.Insert(cData);
        foreach (ArticleData aData in cData.ArticleList)
        {
```

```
            aData.ChannelIndex = cData.Index;
        }
        m_ArticleDAO.InsertAll(cData.ArticleList);

        if (AddChannelEvent != null)
            AddChannelEvent(cData);
    }
}
```

그림 4. R²SS 주소의 동적 디렉토리 구성 방법

2.4 사용자 질의에 따른 R²SS 출처 순위화

R²SS의 기본 목적은 RSS를 통한 콘텐츠의 자동 전달이라고 할 수 있다. 따라서 이를 위해서는 사용자가 원하는 정보가 꾸준히 올라오는 순서대로 RSS 출처의 순위를 정하여 제시해줄 필요성을 가진다. 현재가지 조사해본 바로는 정보 전달을 목적으로 정보 갱신을 고려하여 블로그의 순위를 매기는 검색 도구는 존재하지 않는다.

현재 사용자들이 RSS 출처를 이용하기 위해 차선책으로 사용자는 블로그의 제목과 소개글 대상의 검색이 주로 이루어지므로 정보 갱신 여부와는 무관한 순위가 제시된다. 또한 기존에 제안된 순위와 방식을 사용하는 것에도 무리가 있다. 기존의 유사도에 따른 순위화는 사이트 단위가 아닌 문서 단위의 유사도 계산이 적합하다. 예를 들어, 사용자가 질의한 단어를 다수개 포함한 글이 1개 올라온 RSS 출처와 사용자가 질의한 단어를 한 개씩 포함한 다수개의 글이 올라온 RSS 출처의 경우를 생각해 보면, 단순한 유사도 계산으로 순위를 매기면 전자가 우선 순위를 가질 수도 있게 된다.

따라서 본 논문에서는 사용자에게 우선적으로 추천되어야 할 RSS 출처는 “자료의 갱신주기가 짧고, 사용자가 관심 있어 하는 글이 많이, 그리고 꾸준히 올라올 만한 출처”라 정의하고 이를 위해 사용자 질의 단어를 포함하는 게시물의 수, 자료의 개인률, 자료의 갱신 주기 등을 고려한 RSS 출처 순위화 방법을 제안한다. 본 논문에서 제시된 방법을 적용하기 위해서는 샘플 기간과 샘플 주기 설정을 필요로 한다.

R²SS 출처의 순위를 정하기 위한 첫 번째 기준은 식 (1)과 같다.

$$channel\_rank1 = \sum_{i=1}^{i=m}(k/n)\times w_i \qquad (1)$$

channel_rank1은 질의 단어를 포함한 게시물 수와 질의 단어 위치에 따른 우선순위 값을 의미하며, 수식의 변수에서 k는 질의 단어수, n은 질의 단어 중 해당 게시물과 매칭된 수, m은 샘플 기간 내 질의 단어를 포함한 게시물의 수, 그리고 wi는 질의 단어 위치에 따른 가중치를 의미한다.

첫 번째 기준은 샘플 기간내에서 존재하는 질의한 단어를 포함한 게시물의 양에 기반하여 순위를 매기게 된다. 만약 첫 번째 기준의 결과가 동일한 경우, 식 (2) 와 같은 두 번째 기준을 적용하여 순위를 정한다. 두 번째 기준은 사용자의 정보 갱신의 일관성 여부를 파악하기 위한 것이다.

$$channel\_rank2 = \frac{1}{n}\sum_{i=1}^{n}(k_i - m)^2 \qquad (2)$$

channel_rank2는 식 (1)의 결과가 동일할 경우 두 번째 기준을 적용하여 순위를 정하기 위한 것으로, 수식의 변수에서 n은 샘플링 기간을 샘플링 주기로 나눈 결과 이며, ki는 I번째 주기의 질의 단어를 포함하는 게시물의 수, 그리고 m은 샘플링 기간내 평균 게시물 개수를 의미한다.

예를 들어, 샘플 기간 2달, 샘플 주기 1주로 설정하였고, 블로그1과 블로그2의 샘플 주기 당 질의 단어 포함 게시물의 수가 블로그1(1,2,1,2,1,2,1,2)이고 블로그2(0,4,0,5,0,2,0,1)로 가정한다면, 샘플 기간내 질의 단여 포함 게시물의 수는 같지만 블로그1과 블로그2보다 더 작은 분산 값을 가지므로 더 높은 순위를 가지게 된다.

만약 식 (1)과 식 (2)의 기준이 모두 같을 경우라면, 식 (3)과 같이 샘플 기간 내 질의 단어 포함 게시물의 갱신 주기를 파악하여 갱신 주기가 짧은 것이 더 우선 순위를 가지도록 한다. 본 논문에서는 짧은 기간을 1년 이내로 제안하며, 샘플 주기와 자료 갱신 주기는 모두 일단위로 처리된다.

$$channel\_rank3 = \frac{1}{m-1}\sum_{j=1}^{m-1}(d_{j+1} - d_j) \qquad (3)$$

channel_rank3은 식 (1)과 식 (2)의 결과가 동일할 경

우 세 번째 기준을 적용하여 순위를 정하기 위한 것으로, 수식의 변수에서 m은 샘플 기간 내 질의어 포함 게시물의 개수, 그리고 dj는 j번째 게시물의 갱신일을 의미한다.

실제로 국내 블로그를 대상으로 모아진 RSS 출처 정보에 제안된 순위화 방법을 적용한 결과, 검색 순위 선정 시 대부분 첫 번째와 두 번째 기준에 따라 순위가 정해지는 것을 확인할 수 있었다. 이유는 아직까지 일관성을 가지고 꾸준히 자료를 올리는 RSS 출처의 수가 그만큼 적다는 뜻으로 풀이될 수 있다.

2.5 R²SS 주소 데이터베이스

R²SS 주소 수집은 자동적인 확장방식으로 대표적인 인터넷 자원(예컨대, RSS 또는 ATOM 등) 주소 표현 형태를 이용하여 인터넷(Internet) 상에 있는 웹(예컨대, IPv4에서는 일반 웹, IPv6에서는 전자제품을 포함한 웹 등) 문서 (HTML 파일)에서 RSS 주소를 자동적으로 추출하고, 해당 웹 문서에 있는 링크(link)에서도 같은 방식으로 RSS 주소를 추출하는 방식이다. 즉, 미리 설정된 주요 포털이나 블로그 웹 문서를 시작으로 해서 점차적으로 해당 웹 문서들의 외부로 향하는 링크를 따라 방문하면서 RSS 주소를 자동 추출하거나, RSS 주소를 추출할 웹 문서를 주요 메타 사이트들이 제공해주는 최신 RSS 파일을 주기적으로 방문하면서 이에 들어 있는 링크 주소를 방문하여 RSS 주소를 추출하는 방식이다.

이렇게 수집된 RSS는 기존 데이터베이스에 있는지를 체크한 다음에 없으면, R²SS 주소 데이터베이스에 새로 추가한다. R²SS 주소 데이터베이스 구조는 [표 2]와 같다.

표 2. R²SS 주소 데이터베이스 구조

RSS 주소	마지막 방문날짜	갱신 주기
http://mensa-barbie.blogspot.com/rss.xml	2009-07-31	0
http://thecaveman.blogspot.com/rss.xml	2009-07-31	0

3. R²SS 데이터 수집 처리

3.1 블로그 내에서 본문 추출

사용자가 원하는 RSS 출처를 찾았으면, 이 주소를 R²SS 데이터 수집 처리와 같은 소프트웨어에 등록함으로써 신규 정보를 자동으로 전달 받을 수 있게 된다. 그 문서의 구조는 [그림 5]와 같다.

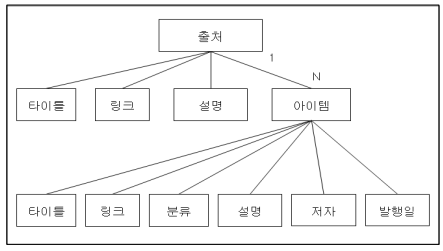


그림 5. R²SS 문서의 구조

[그림 5]에서 R²SS 문서의 처리를 위해 R²SS 데이터 수집 처리를 구현하였으며, 등록된 출처 정보를 이용하여 URL 객체를 생성한다. 생성된 URL 객체를 이용하여 RSS 파일을 가져와 R²SS Parser를 이용하여 파싱한 후, 출처 객체로 만들어 필요한 엘리먼트를 추출한다. 만약 기존 출처 객체가 파일로 존재할 경우, 출처 객체를 복원해 내어 현재 출처와 비교하는 과정을 통해 사이트의 신규 자료 여부가 판단된다.

3.2 신규 데이터의 분류

[그림 5]에서와 같이 R²SS에 포함된 아이템의 하위 타이틀과 설명 엘리먼트에는 신규 자료의 제목과 내용 정보가 위치한다. 따라서 이 정보에서 색인어를 추출하고 R²SS 데이터 수집 처리가 가지고 있는 분야별 색인어 집단의 유사도 비교를 통해 데이터의 분류를 하게 된다.

(1) 유사도 계산

R²SS 문서에 포함된 신규 정보의 제목과 내용에서 어절 단위 분리, 조사, 불용어 제거 작업을 통해 문서 내 존재하는 모든 단어들을 추출하고 해당 단어의 빈도수를 기억한다. 추출된 모든 단어와 단어 빈도수는 웹 콘텐츠 분류 시에 사용된다.

이력 문서 내에서 추출된 단어들은 분야별 색인어 유

사도 계산을 통하여, 유사도가 가장 높은 분야에 분류된다. 유사도 계산을 통하여 유사도 계산식은 식 (4) 와 같이 간략화시켜 적용한다.

$$S\_Comp(C,d) = \frac{\sum_{i=1}^{N_c}(freq_{i,d} / \max_{i,d})}{N_c} \quad (4)$$

S_Comp(C,d)는 R²SS 문서에 포함될 신규 데이터 분류를 위한 것으로, 수식의 변수에서 Nc는 c분야 색인어 집단에서의 총 색인어 수, freqi,d는 입력문서 d에서 매칭되는 단어 i의 빈도수, 그리고 maxi,d는 문서 d에서 가장 자주 등장하는 단어의 빈도수를 의미한다.

(2) 예외 처리

R²SS 문서를 분류 시에 데이터의 신뢰도를 높이기 위해 스팸 분류와 데이터 중복 및 펌 문서 체크, 그리고 자동 분류 기능을 추가하였다.

스팸 분류는 서포트 벡터 머신(Support Vector Machine, SVM)을 활용한 자동분류기를 이용하여 개발하였다. 서포트 벡터 머신은 통계 분류와 회귀 분석을 쓰는 지도 학습 방법을 가리키는 말이다. 이는 커널 트릭을 써서 비선형 분류 문제에 선형 분류의 테크닉을 적용한다.

서포트 벡터 머신은 현재 알려져 있는 많은 학습법 중에서 가장 인식 성능이 뛰어난 학습 모델의 하나이다. 서포트 벡터 머신이 뛰어난 인식 성능을 발휘할 수 있는 이유는 미 학습 데이터에 대해서 높은 식별 성능을 얻기 위한 방법이 있는 것이다. 서포트 벡터 머신은 선형 문턱 소자를 이용하고, 2 클래스의 패턴 식별기를 구성하는 기법이다. 훈련 샘플로부터 「마진 최대화」라고 하는 기준으로 선형 문턱 소자의 파라미터를 학습한다.

중복 및 펌 문서 체크 기능은 블로그에서 가장 큰 이슈가 펌 글 즉, 중복된 글을 어떻게 체크하는가이다. 본 논문에서 구현한 중복 문서 체크 모듈은 80% 이상 비슷한 문서이면 중복이라고 판단하고, 원문은 작성일자가 가장 이른 것을 기준으로 한다. 댓글은 사람들이 특정 포스트에 얼마나 관심이 많은지 보여주는 객관적인 지표임으로, 따로 분리를 하여 관리 할 필요가 있다. 현재는 댓글의 개수만 세고 댓글의 내용은 저장 하고 있지 않다.

자동 분류 기능은 효율적으로 모니터링을 하기위해서 필수이다. 분류는 사람들이 관심이 있는 분야를 관찰하여 대분류 6개 (IT.과학, 시사, 연예, 일상, 게임, 스포츠) 그리고 25개 소분류로 구성 하였다. 자동분류 알고리즘은 스팸분류와 같이 서포트 벡터 머신을 이용한 자동분류 엔진을 활용하여 구현하였다.

3.3 R²SS 문서 데이터베이스

R²SS 데이터수집 처리는 R²SS 주소 데이터베이스에 미리 저장된 RSS 주소들에 해당하는 RSS 파일들을 제공받아 각 RSS 파일이 제공해주는 링크정보를 이용하여 웹 문서 데이터들을 수집하는 기능을 수행한다.

즉, R²SS 데이터수집 처리는 R²SS 주소 데이터베이스와 연동되어 R²SS 주소 수집 처리에 의해 수집 저장된 R²SS 주소 목록을 주기적으로 제공받아 각 RSS 주소를 방문하면서 해당 RSS 파일을 다운로드(Download)받은 후, 각 RSS 파일이 제공해주는 RSS 정보들(예컨대, 제목(title), 링크(link), 요약설명(description), 카테고리(category) 및 등록날짜(publication date) 정보 등) 중 소스 링크정보에 존재하는 링크(link)를 방문하여 해당 웹 문서 데이터를 수집하고 이를 R²SS 데이터 데이터베이스에 전송한다. R²SS 데이터 수집 처리에 의해 생성된 데이터베이스 구조는 [표 3]과 같다.

표 3. R²SS 문서 데이터베이스 구조

제목	링크	요약설명	카테고리	등록날짜
개발자들의 아카데미상	http://agile.egloos.com/323897	IT업계에도 노벨상과 아카데미상이 있다. 튜링상을 노벨상에 비유할 수 있고, 졸트상을 아카데미상.....	IT	2007-05-26

이때, 상기 소스 링크정보에 존재하는 링크 방문 시 R²SS 문서 데이터베이스에 미리 저장된 RSS 파일 목록과 이미 다운로드(Download)받은 RSS 파일을 비교하여 RSS 파일 내용 중에서 갱신된 RSS 정보의 소스 링크정보에 존재하는 링크를 방문하여 수집함이 바람

직하다.

3.4 R²SS 데이터 분석 처리

R²SS 데이터 분석 처리는 R²SS 주소수집 처리와 R²SS 데이터수집 처리로부터 수집된 수많은 RSS 주소들, 각 주소간의 링크 관계, 각 사용자 정보들 및 각 사용자가 요청한 정보별로 형태소들을 데이터베이스화하여 저장 및 관리하는 기능을 수행한다.

3.5 사용자 R²SS 문서 제공

R²SS 데이터 분석 처리는 R²SS 데이터 DB 에서 관리하는 형태소 중에서 R²SS 주소들에 해당하는 R²SS 파일들을 제공받아 각 R²SS 주소들을 방문하면서 해당 R²SS 파일을 다운로드 받은 후, 각 R²SS 파일의 R²SS 정보들 (제목(title), 링크(link), 요약설명(description), 카테고리(category) 및 등록날짜(publication date) 정보 등) 중 소스 링크정보에 존재하는 링크를 방문하여 해당 웹 문서 데이터를 수집하고 이를 R²SS 정보제공 DB에 저장하는 기능을 수행한다.

또한 R²SS 데이터 DB에 저장된 R²SS 파일 목록과 다운로드 받은 R²SS 파일을 비교하여 R²SS 파일 내용 중에서 [그림 6]과 같이 갱신된 R²SS 정보의 소스 링크 정보에 존재하는 링크를 방문하여 해당 웹 문서 데이터를 수집하는 기능을 수행한다.

R²SS 데이터 분석 처리는 R²SS 데이터 수집 처리 엔진에서 수집하여 R²SS 데이터 DB에 저장된 각 R²SS 파일에 대한 웹 문서 데이터를 제공받는다. 또한 형태소 분석모듈을 통해 형태소를 분석하여 R²SS 데이터 DB에 저장된 각 사용자 요청정보별 형태소들과 비교한 후, 각 사용자 요청정보와 동일한 범주에 포함되는지의 여부를 판단하는 기능을 수행한다.

이를 구체적으로 설명하면 R²SS 데이터 DB의 형태소 DB에 저장되어 있는 형태소명 들과 현재 분석중인 웹 문서 데이터를 형태소 분석한 형태소명 들을 비교하여, 현재 웹 문서 데이터가 포함하고 있는 형태소 ID를 추출한다.

사용자 R²SS 문서 제공은 R²SS 데이터 수집 처리에서 형태소 분석된 웹 문서 데이터가 각 사용자 요청정보와 동일한 범주에 포함되면, 해당 웹 문서 데이터에 대한 R²SS 파일의 R²SS 정보들을 제공받아 해당 사용자 요청정보에 대한 R²SS 문서 정보를 생성 및 저장하는 기능을 수행한다.

```
<item>
   <title><![CDATA[개발자들의 아카데미 상 ]]>
   </title>
   <link>http://agile.egloos.com/3238987</link>
   <guid>http://agile.egloos.com/3238987</guid>
   <description>
   <![CDATA[
IT업계에도 노벨상과 아카데미상이 있습니다.
류링상을 노벨상에 비유할 수 있고, 졸트상을
아카데미상에  비유할 수 있습니다.
오늘은 개발자들에게 아카데미상을 보는 기대감과
즐거움을 주는졸트상에 대해 이야기를 해보겠습니다.
졸트상은 1990년부터 .....
<a herf="http://agile.egloos.com/3238987">
<font style="font-size:11px;">글 전체보기
</font></a>]]>
   </description>
   <category>미분류</category>
   <pubDate>Mon, 23 Nov 2008 10:34:50
GMT</pubDate>
</item>
```

그림 6. 적용된 R²SS 파일의 소스

Ⅳ. 실험 및 성능 평가

본 장에서는 웹 2.0기반의 대용량 데이터 검색을 위한 R²SS 웹 크롤러 시스템의 성능을 평가하기 위하여 메타 블로그 사이트와의 성능 비교, R²SS 출처 순위화의 실험과 성능 평가, 그리고 R²SS 데이터 형태소 분석 처리의 실험과 성능 평가를 실시한다.

1. 메타 블로그 사이트와의 성능 비교

1.1 실험 환경

본 논문에서 구현한 R²SS 기반의 정보검색 시스템은 다수의 블로그 사이트로부터 정보를 수집하여 한 곳에서 관리한다는 측면에서 일종의 메타 블로그 사이트와 유사하다. 따라서 국내에 존재하는 메타 블로그 사이트(AllBlog, BLOZINE, BLOG KOREA)와의 블로그 정보 관리 기능을 비교한다.

1.2 성능 평가

국내에 존재하는 메타 블로그 사이트의 블로그 정보

관리 기능을 비교한 결과는 [표 4]와 같다. RSS 채널 수집 기능을 보면, 현재의 메타 블로그 사이트의 경우 가입자가 스스로 자신이 관리하는 블로그의 RSS 채널 주소를 입력함으로써 수집되는 반면, 제안하는 시스템은 RSS 채널만을 탐색하는 웹 크롤러를 통해 자동으로 수집할 수 있다. 블로그 검색 도구의 경우 아직까지 블로그의 정보 갱신 정도를 바탕으로 검색 순위를 지정하는 검색도구는 존재하지 않는다. 블로그 디렉토리 기능은 BLOZINE 사이트에서도 지원하지만, 사용자가 가입 시 설정한 디렉토리로 고정적으로 유지되므로, 사용자가 디렉토리 접근을 통해 블로그 검색 시 효율이 떨어지는 단점이 있다.

표 4. 메타 블로그 사이트와 기능 비교

비교대상 / 비교항목	제안 시스템	AllBlog	BLOZINE	BLOG KOREA
RSS 채널의 자동 수집	○	×	×	×
RSS를 이용한 주기적인 블로그 정보 수집	○	○	○	○
블로그 검색도구	○	△	△	△
블로그 디렉토리 제공	○	×	△	×
역 RSS를 이용한 관련 정보 구성	○	×	×	×

기존 RSS 제품은 정보수집 범위가 한정되어 있는데 반해, 본 R^2SS 기반의 정보검색 시스템은 역 RSS 정보검색.수집 방식으로 전체 인터넷 망을 대상으로 광범위한 정보 검색이 가능하다. 단순히 정보를 많이 모으는 것뿐만 아니라 그 정보를 대용량 데이터 자동 분류 기법을 사용하여 정보를 조직화 하고 분류하여 정보의 활용도를 극대화 시킨다.

기존의 국내 메타 블로그는 대략 20만 ~ 30만 정도의 RSS 주소만 수집 되어 있으며, 기존 방식으로 새 글을 수집하기 위하여 하루에서 심지어 며칠씩 걸리는 반면, 본 연구를 통해 개발된 R^2SS 기반의 정보검색 시스템은 수집 시간이 몇 시간 내외이며 파워 블로그나 뉴스 같이 자주 업데이트 되는 사이트는 수집 시간이 몇 십 분 심지어 10분 이내로 이루어지고 있다.

2. R^2SS 출처 순위화의 성능 평가

2.1 실험 환경

본 논문에서 제시한 사용자 질의에 대한 R^2SS 출처 순위화를 사용한 경우와 국내의 블로그 서비스를 제공하는 포털 사이트 중 엠파스(www.empas.com)에 존재하는 블로그 검색 도구와 비교 실험한다. 엠파스)에서 제공되는 블로그 검색 도구는 블로그 제목(소개글 포함) 대상 검색과 블로그 카테고리 검색, 블로그 신규 게시물 검색을 제공한다. 다음, 파란, 하나포스, 코리아닷컴, 메이저 등도 동일한 검색 도구를 지원한다.

2009년 9월 13일 기준으로 검색하였으며, 각 검색 도구의 상위 랭크된 10개 블로그의 RSS 주소를 취하여 사용자가 질의한 검색어를 포함한 신규 자료의 갱신 정도를 비교하였다. 검색어는 '윈도우' 이며, 영문 'Windows'도 함께 포함하여 검색한다. 제안된 시스템의 R^2SS 채널 순위화 방법을 이용하기 위해서는 샘플 주기와 기간 설정이 필요한데, 여기서는 샘플 주기는 14일로 선택하고 2009년 8월 17일부터 2009년 9월 13일까지(27일)을 샘플링 기간으로 설정하였다.

2.2 성능 평가

본 논문에서 제시한 사용자 질의에 대한 R^2SS 출처 순위화를 사용한 경우와 국내의 블로그 서비스를 제공하는 포털 사이트 중 엠파스에 존재하는 블로그 검색 도구와 비교 실험을 해본 결과 [그림 7[과 같은 결과를 얻을 수 있었다.

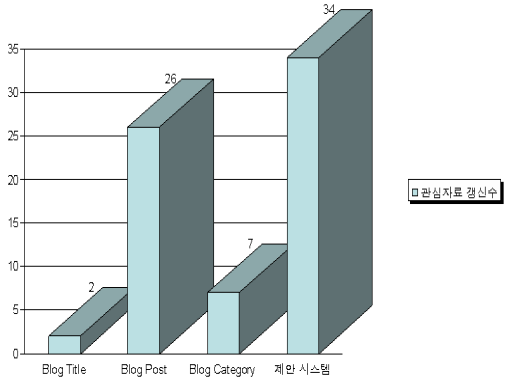


그림 7. R^2SS 출처 순위화의 성능 비교

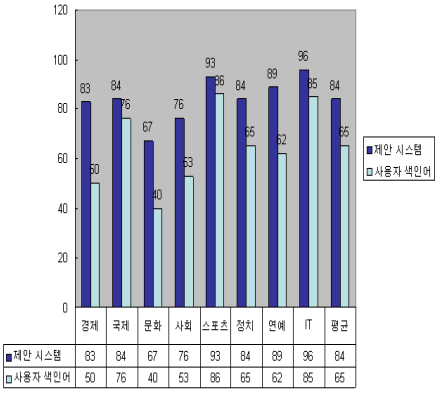
[그림 7]의 결과에서 알 수 있듯이, 실제 포털 사이트에서 제공하는 검색 결과를 사용한 경우보다 제안된 시스템의 검색 결과를 사용한 경우의 관심 자료의 갱신률이 더 뛰어난 것으로 나왔다. 이는 기존 블로그 검색엔진을 통해서는 블로그의 사이트 제목과 소개 글에 바탕을 둔 결과가 나오기 때문에 앞으로의 갱신성에 대한 고려는 없었다. 하지만 제안된 시스템은 사용자 질의어에 기반한 자료의 수, 갱신 주기, 갱신률의 분산도 등을 모두 고려하였기 때문에 기존 결과에 비해 우수한 성능을 보일 수 있었다.

3. R²SS 색인어 처리에 따른 성능 평가

3.1 실험 환경

(주)에버아이티에 재직 중인 근로자 5명을 대상으로 각 분야별로 색인어를 추출 작업을 실시하였다. 실제 사용자가 해당 분야에 관련된 색인어를 선정할 때는 개인의 경험에 비추어 각 분야를 대표하는 명사를 선정하는 경향을 보였다. 그리고 선정 개수는 최소 5개에서 최고 20개였다. 실험에 참가한 5명이 선정한 색인어를 바탕으로 각 분야별 10~20개 정도의 색인어를 선별적으로 부여하였다. 제안된 시스템의 빈도수 기준의 색인어 추출 방법과 사용자의 명시적 색인어 선정 방법으로 비교하였다.

3.2 성능 평가



	경제	국제	문화	사회	스포츠	정치	연예	IT	평균
제안 시스템	83	84	67	76	93	84	89	96	84
사용자 색인어	50	76	40	53	86	65	62	85	65

그림 8. R²SS 색인어 처리에 따른 성능 비교 (단위 : %)

제안하는 시스템은 사용자가 만든 분야에 초기 문서를 지정해 놓으면, 초기 문서로부터 중요 빈도수 기준과 분야별 중복 색인어 제거 기준을 가지고 분야별 색인어 집단을 자동으로 추출해낸다. 이의 효율성을 알아보기 위해 사용자의 분야별 명시적 색인어 추출 경우와 비교 실험을 해보았다.

[그림 8]에서와 같이 제안된 방법이 평균 19% 정도 성능 향상을 보였다. 이 결과는 실제로 사용자가 임의로 설정한 색인어는 그 수가 너무 적어 분류시 미분류되는 경우도 다수 발생하였으며, 또한 색인어가 분야별 중복 색인어의 제거 기준 없이 생성되었기 때문에 잘못 분류되는 경우도 다수 발생하였다.

V. 결론

기존의 RSS 리더기가 가지는 문제점을 해결하기 위한 것 중 하나가 메타 블로그(Meta Blog)로서 이는 여러 사람이 수동으로 입력한 RSS 주소를 공유해서 다양한 RSS 주소로부터 콘텐츠를 가져와 사용자에게 보여주거나 검색할 수 있도록 하는 것이다. 이렇게 RSS 주소가 좀더 많아지기는 하였지만 여전히 사용자가 수동으로 입력한 극소수의 RSS 주소에 의존하고 있다. 또한, 사용자는 다양한 사람들이 입력한 RSS 주소로부터 정보를 받아 보면서 그 중에 자신이 원하는 정보를 선택해야 하는 수고를 해야 한다.

본 논문에서는 이러한 문제점을 해결하기 위하여 사용자가 RSS 주소를 입력하여 제한된 정보를 받아 보는 방식이 아니라 사용자는 단순히 자신이 원하는 정보를 입력만 하면, 자동화된 RSS 주소수집서버가 수집한 수많은 RSS 주소들로부터 실시간으로 수집하는 RSS 정보들 중에서 사용자가 원하는 정보에 대한 역 RSS 문서 정보를 제공하여 RSS의 사용 용이성 한계와 제공되는 정보 범위의 한계를 극복할 수 있도록 한 R²SS 기반 지능형 정보 검색 시스템을 제안하였다.

본 논문에서 제안하는 기법들은 다른 기법들과의 비교 실험을 수행하여 실제 성능이 우수함을 증명하였다. 성능 평가는 기존 메타 블로그와의 RSS 채널의 자동

수집, 주기적인 정보 수집 등의 성능 비교, 제안한 시스템의 RSS 채널 순위화 방법에 따른 RSS 채널 추천의 성능 비교, 색인어에 따른 형태소 분석을 통한 분야별 색인어 검색의 성능 비교, 그리고 대용량 처리에 따른 성능 비교를 실시하였다. 그 적용 사례로 블로그 코리아, AllBlog, BLOZINE 블로그 사이트와 대용량 처리를 위한 벤치마크 애플리케이션을 대상으로 실험하고 그 결과를 분석하였다.

성능 평가 결과, 첫 번째로 메타 블로그 사이트와의 성능 비교에서 기능면에서 자동 수집, 주기적인 블로그 정보 수집, 블로그 검색 도구, 블로그 디렉토리 제공, 역 RSS 기법을 적용한 정보 수집에서 우수하였으며, 속도면에서 기존의 국내 메타블 로그는 대략 20만 ~30만 정도의 RSS 주소만 수집이 돼있으면 기존 방식으로 새글을 수집하기 위하여 하루에서 심지어 몇일씩 걸리는 반면, 본 연구를 통해 개발된 R^2SS 기반의 정보검색 시스템은 수집 시간이 몇시간 내외이며 파워블로그나 뉴스같이 자주 업데이트 되는 사이트는 수집시간이 몇십분 심지어 10분 이내로 이루어지고 있다.

둘째로 R^2SS 출처 순위화의 성능 평가에서 실제 포털 사이트에서 제공하는 검색 결과를 사용한 경우보다 제안된 시스템의 검색 결과를 사용한 경우의 자료 갱신률이 더 뛰어난 것으로 나왔다. 이는 기존 블로그 검색 엔진을 통해서는 블로그의 사이트 제목과 소개글에 바탕을 둔 결과가 나오기 때문에 앞으로의 갱신성에 대한 고려는 없었다. 하지만 제안된 시스템은 사용자 질의어에 기반한 자료의 수, 갱신 주기, 갱신률의 분산도 등을 모두 고려하였기 때문에 기존 결과에 비해 우수한 성능을 보일 수 있었다.

셋째로 R^2SS 색인어에 따른 성능 평가에서 제안된 방법이 평균 18% 정도 성능 향상을 보였다. 이 결과는 실제로 사용자가 임의로 설정한 색인어는 그 수가 너무 적어 분류시 미분류 되는 경우도 다수 발생하였으며, 또한 색인어가 분야별 중복 색인어의 제거 기준 없이 생성되었기 때문에 잘못 분류되는 경우도 다수 발생하였다.

본 논문의 기대효과는 사용자가 단순히 자신이 원하는 정보를 입력만 하면, 자동 수집된 수많은 RSS 주소들로부터 실시간으로 수집하는 RSS 규격 문서들 중에서 사용자가 원하는 RSS 규격 문서에 대한 RSS 정보만을 제공해줌으로써, 사용자는 수많은 정보를 찾아서 그 중 원하는 정보만 추려서 제공해주는 개인 비서를 두게 되는 효과를 얻게 되어서 양질의 정보를 찾아 헤매는 시간을 획기적으로 줄일 수 있다.

향후의 과제로는 대용량 콘텐츠의 자동 수집을 위한 지능형 정보검색시스템을 역 RSS 기술과 웹 크롤러 기술을 적용하기 위하여 한, 중, 일, 영어에서의 각각 많은 RSS 주소(각각 140만, 50만, 30만, 50만)를 확보하여 이러한 RSS 주소를 통하여 대량의 문서를 수집하는데 있다.

참 고 문 헌

[1] 홍석주, 박영배, “대용량 콘텐츠를 위한 역 RSS 웹 크롤러 설계”, 한국통신학회 논문지 제34권 제2호, 2009(2).

[2] 홍석주, 박영배, “대용량 콘텐츠를 위한 역 RSS 웹 크롤러 구현”, 한국통신학회 논문지 제34권, 제3호, 2009(4).

[3] 장은영, “개별화된 웹 미디어를 이용한 학습 커뮤니티 환경 설계 및 구현”, 한국교원대학교 석사학위논문, 2005.

[4] 강성후, “XML을 활용한 rss 리더기의 설계 및 구현”. 부산외국어대학교 석사학위논문, 2005.

[5] 김중태, *나는 블로그가 좋다*, 이비컴, pp 50-250, 2004.

[6] 김법묵, “동기적 상호작용 증진을 위한 블로그 기반 협동학습 시스템의 설계 및 구현”, 한국교원대학교 석사학위논문, 2005.

[7] M. Bornemann, R. V. Nieuwpoort, and T. Kielmann., "MPJ/Ibis: a flexible and efficient message passing platform for Java," In Proceedings of 12th European PVM/MPI Users' Group Meeting, pp.217-224, 2005(9).

[8] A. Bouteiller, F. Cappello, T. Herault, G. Krawezik, P. Lemarinier, and F. Magniette, "MPICH-V2: a Fault Tolerant MPI for Volatile Nodes based on

Pessimistic Sender Based Message Logging," In Proc. of the 15th International Conference on High Performance Networking and Computing(SC2003), November 2003.

[9] R. Metkowski and P. Bala, "Parallel Computing in Java: Looking for the Most Effective RMI Implementation for Clusters," Lecture Notes in Computer Science, Springer-Verlag Berlin, Vol.3911, pp.272-277, 2006.

[10] C. Nester, M. Phillippsen, and B. Haumacher, "A more efficient RMI for java," In Proc. of the ACM Java Grande Conference, pp.152-159, 1999(6).

[11] M. Philippsen, B. Haumacher, and C. Nester, "More efficient serialization and RMI for Java," Concurrency: Practice and Experience, Vol.12, No.7, pp.495-518, 2000.

[12] RSS 2.0 Specification, http://blogs.law.harvard.edu/tech/rss

[13] Weihong Huang, "Enabling Context-Aware Agents to Understand Semantic Resources on the WWW and The Semantic Web," Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence, pp.138-144. 2004.

[14] Y. Nakano, "A proposal of RSS WebCrawler model of product information," Proc. of the 2005 International Conference on Active Media Technology, pp.147-151. 2005.

[15] De Sutter, R. "Enhancing RSS Feeds: Eliminating overhead through Binary Encoding," Third International Conference on Information Technology and Applications, pp.520-525. 2005.

[16] F. Menczer, "Evaluating topic-driven Web crawlers," Proc. 24th annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp.241-149. 2001.

[17] Cautam Pant, "Topical Crawling for Business Intelligence," Proc. of ECDL 2003, pp.233-244, 2003.

[18] B. Yuwono, "Search and ranking algorithmms for lacating resources on World Wide Web," Proc. of the Int. Conf. on Data Engineering (ICDE), pp.164-171, 1996.

[19] Jon Kleiboemer, "Authoritative sources in a hyperlinked environ -ment," Proc. of the 9th ACM-SIAM symposium on Discrete Algorithms, pp.668-677, 1998.

[20] L. Page, "The Pagerank citation ranking: Bring Order to the Web(Tech. Rep.)," Sranford Digital Library Technologies Project. 1998.

저 자 소 개

홍 석 주(Seok-Joo Hong) 정회원

- 1988년 2월 : 명지대학교 전자계산학과 졸업(공학사)
- 1998년 8월 : 명지대학교 컴퓨터공학과(공학석사)
- 2001년 8월 : 명지대학교 컴퓨터공학과 박사수료
- 방송정보기술사, ISO 국제심사원
- 경희사이버대학교 정보통신학과 겸임교수

<관심분야> : web computing, ITS, multimedia Database

박 영 배(Young-Bae Park) 정회원

- 1993년 2월 : 서울대학교 대학원 컴퓨터공학과(공학박사)
- 1990년 ~ 1992년 : 명지대학교 전산소장
- 1997년 ~ 2001년 : 명지대학교 산업대학원장
- 1980년 3월 ~ 현재 : 명지대학교 컴퓨터공학과 교수

<관심분야> : Mobile DB, Spatial DB, 한국어정보처리, Large Fingerprint DB, Web computing