사용자 관심분야에 따른 RSS 채널 추천 시스템

*김준일, 이영석, 조정원, 최병욱
*한양대학교 정보통신공학과
한양대학교 전자통신컴퓨터공학과
제주대학교 컴퓨터교육과
한양대학교 정보통신학부
e-mail : {jjunil, yslee38, bigcho, buchoi}@mlab.hanyang.ac.kr

RSS Channel Recommendation System based on Interesting Field

*Jun-Il Kim, YoungSeok Lee, Jungwon Cho, Byung-Uk Choi
*Department of Information and Communications, Hanyang University,
Department of Electrical and Computer Engineering, Hanyang University,
Department of Computer Education, Cheju National University,
College of Information and Communications, Hanyang University

Abstract

We propose the RSS Channel retrieval system to activate the blog information transmission. The system consists of a web crawler and blog DB. Web Crawler moves in limited breath first searching method and it collects the RSS Channel Address. Blog DB renews information using RSS. The user could be recommended the RSS Channel using the various query.

I. 서 론

RSS(Really Simple Syndication)는 사이트의 최근 추가되거나 변경된 페이지들에 대한 요약 정보를 제공하는 기술로서, 사용자가 신규 정보를 찾기 위해 반복적으로 사이트에 접근하는 과정을 줄여줄 수 있다. 2004년 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 뉴스 Aggregator(Reader)를 사용하고 있다[1]. 사용자는 RSS Aggregator에 해당 RSS 채널의 주소만 기억해 두면, 나중에는 이 파일에 요약된 내용만 보고 중간 과정 없이 직접 해당 페이지로 바로 접근할 수 있게 되므로 방문자들의 수고가 크게 줄어든다. 이러한 RSS는 1인 미디어 플랫폼인 블로그의 확산과 함께 늘어나고 있는 추세이다.

하지만, 현재의 시스템은 사용자가 직접 블로그를 돌아다니며 RSS 채널의 주소를 찾아 일일이 등록하는 작업을 통해 이루어지고 있다. 또한 아직까지 일반 사용자들에게는 RSS 문서 구분과 링크위치 파악에 어려운 점이 존재한다. 대부분 블로그 서비스를 제공하는 포탈 사이트에는 블로그 검색툴이 존재한다. 하지만, 이것은 해당 포탈의 블로그로만 검색이 제한되며 블로그 제목, 소개글 정도로만 블로그를 검색해 낸다. 또한 글이 올라오지 않는 비어있는 블로그도 다수 검색되므로 효율적이지 못하다. 이외에 게시물 단위의 검색도 가능하다. 하지만 게시물 하나가 해당 블로그를 대표한다고 보기는 어렵다. 따라서 사용자는 스스로 블로그의 여러 글을 살펴보며 관심 블로그 여부를 파악하는 수고를 해야 한다.

따라서 본 논문에서는 하나의 포탈 사이트에 국한되지 않으며 검색이 가능하도록, 블로그 서비스를 제공하는 여러 포탈 사이트로부터 RSS 채널을 자동으로 수집하는 RSS 채널 탐색 크롤러와 RSS를 이용하여 블로그의 정보 DB를 구성하는 시스템을 제안한다. 제안된 시스템을 통해 사용자는 다양한 질의를 통해 블로그의 RSS 채널을 직접 검색함으로써 RSS Aggregator 사용 시 등록할만한 채널 주소를 찾는데 도움을 얻을 수 있다.

II. 관련연구

2.1 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 웹 문서 내부에 포한 된 Hyperlink를 따라서 이동하면서 웹상의 정보를 수집하는 일종의 소프트웨어이다. 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[2].

웹 크롤러는 기본적으로 DFS(Depth First Searching), BFS(Breath First Searching) 방식으로 웹의 모든 문서를 탐색한다. 만약 크롤러가 탐색하고자 하는 대상이 웹의 전체가 아닌, CiteSeer와 같이 특정 관심 분야에 대한 웹 문서만 검색을 하는 경우, 문서에 나와 있는 모든 링크를 대상으로 움직일 필요는 없다. 이 경우 특정 토픽과 문서와의 연관성에 따라 탐색할 링크의 수를 줄여나간다. 이러한 크롤러를 Focused Crawler 또는 Topical Crawler라고 한다[3]. Naive Best-First Crawler는 웹 문서를 단어의 집합으로 보고 페이지와 주어진 특정 주제와의 연관성을 Cosine Similarity로 계산해낸다. 이에 대한 괜찮은 성능은 이전 연구에서 밝혀져 있다[4]. 이외에도 HTML 페이지의 트리구조나 Document Object Model(DOM)을 이용하여 링크의 구문을 파악하여 이동하는 DOM 크롤러 등이 있다[5].

본 논문에서 사용된 RSS 채널 탐색 크롤러는 기존 연구에서 밝혀진 대로 DFS보다 효과적으로 웹을 탐색할 수 있는 BFS 방식으로 구현되었다. 또한 탐색 도메인을 RSS 채널 링크가 주로 존재하는 블로그 포탈 사이트로 국한시킴으로써 채널 탐색 효과를 높이도록 하였다.

2.2 블로그 정보 검색

1인 미디어인 블로그의 증가는 활용할 수 있는 정보의 양을 대폭 늘었다. 이러한 블로그들의 많은 정보를 효율적으로 사용자들에게 제시하기 위해 각종 메타 블로그 사이트들이 존재한다. 국내의 대표적 메타 블로그 사이트로는 블로그코리아[7], 올블로그[8], 블로진[9] 등이 존재한다. 하지만 아직까지 이러한 블로그 사이트들은 가입자의 직접 등록에 의해서만 블로그 정보가 수집되며, 가입시 블로그 주소와 함께 RSS 채널 주소까지 명시적으로 적어주어야만 정보를 모아올 수 있다. 또한 현재의 메타 블로그 사이트의 검색은 블로그 제목, 소개글로만 국한되어 검색된다. 본 논문에서 제안하는 시스템은 RSS 채널 정보 수집이 사용자의 인터렉션 없이 RSS 채널 탐색 크롤러를 통해 자동화 되며 RSS 문서를 통해 글의 작성자, 작성시간 등을 추출하여 저장함으로써, 사용자가 블로그 검색 시 다양한 질의가 가능하도록 지원한다.

III. 시스템 설계

시스템은 <그림 1>과 같이 RSS 채널을 획득하기 위한 RSS 채널 탐색 크롤러 부분과 탐색된 RSS 채널을 분석하여 DB에 저장하는 모듈로 구성된다. 획득된 RSS 채널은 이후 사용자의 질의어에 따라 RSS 채널을 추천하게 된다.

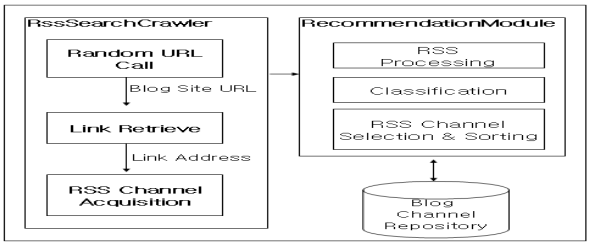

그림 1. 시스템 구성도

3.1 RSS 채널 탐색

RSS 채널 탐색 크롤러는 블로그를 대상으로 블로그에 존재하는 RSS 채널 주소를 찾아 모아온다. 이러한 RSS 채널 주소를 찾기 위한 크롤러 구현을 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 채널 주소 링크는 대개 블로그 메인 페이지에 존재한다. 따라서 블로그에서 RSS 채널 주소를 찾기 위한 링크 탐색 범위는 블로그 메인 페이지의 링크 개수로 제한되어야 한다.

이렇게 고려된 사항을 바탕으로 RSS 채널 수집 효과를 늘이기 위하여, 본 논문에서 제시된 크롤러는 블로그 포탈 사이트의 랜덤 블로그 접근 모듈을 사용한다. 해당 모듈의 호출을 통해 크롤러는 블로그 주소를 획득하며, 블로그 메인 페이지에 존재하는 링크의 수로 탐색 큐의 사이즈를 제한 후 Breath-First 방식으로 링크들을 조사한다.

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

본 시스템에서 제안하는 RSS 채널을 획득하기 위한 과정은 <그림 2>와 같다.

이때 크롤러는 각 블로그 사이트 메인 페이지에 존재하는 태그들을 조사하기 위해 다음과 같은 '정규표현식(Regular Expression)'을 이용한다[6].

- 프레임 태그 패턴 :
  <frame₩₩s+(.*?)₩₩s+src₩₩s*=₩₩s*₩"?(.*?)[₩"|>]
- 링크 태그 패턴 :
  <a₩₩s+href₩₩s*=₩₩s*₩"?(.*?)[₩"|>]
- RSS 채널 여부 판단을 위한 패턴 :
  ^<channel

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 링크주소를 획득한다. 크롤러는 획득된 링크주소들에 접근하여 RSS채널이 가져야 할 태그를 가진 여부를 확인 후 RSS채널이라 판단되면, 기존에 이미 탐색된 채널인지의 비교를 통해, 신규 채널일 경우 해당 RSS 채널 주소를 저장한다.

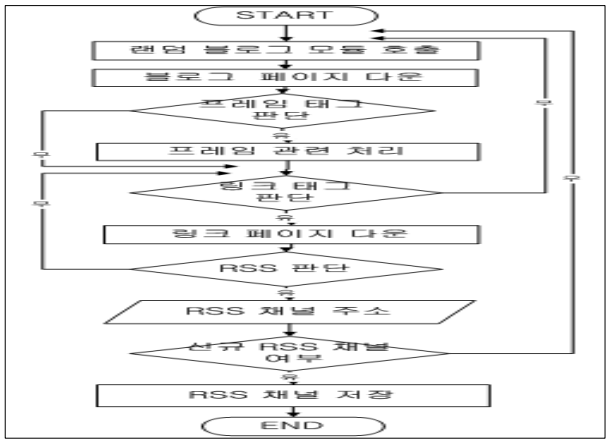

그림 2. RSS 채널 탐색 흐름도

3.2 RSS 채널 관리

습득된 RSS 채널은, 추천 후보 채널과 비추천 채널로 구분되어 관리된다. 여기서 비추천이란 <그림3>과 같이 현재 블로그에 올라온 글이 없는 경우를 말한다. 이는 RSS 문서에 포함된 Item 여부로 확인할 수 있다.

```
<?xml version="1.0" encoding="euc-kr" ?>
- <rss version="2.0" xmlns:blogChannel="http://backend.userland.com/blogChannelModule">
  - <channel>
      <title>참조</title>
      <link>http://blog.hankooki.com/dplee6741</link>
      <description>모든것을 새롭게 시작합니다</description>
      <language>ko</language>
      <copyright>Copyright, 떠판</copyright>
      <managingEditor>dplee6741</managingEditor>
      <lastBuildDate>2004-06-09 오전 10:16:11</lastBuildDate>
      <generator>PersonalDB Blog XML</generator>
      <ttl>60</ttl>
    </channel>
</rss>
```

그림 3. 신규 정보를 포함하지 않은 RSS 문서

비추천 채널이 아닌 나머지 채널은 추천 후보 채널로 배정되며, 이에 해당되는 채널들은 실제 채널 주소에 존재하는 RSS 문서로부터 정보를 추출하여 DB에 저장한다. RSS 문서로부터 추출 가능한 정보는 <그림 4>와 같다. 추천 후보 채널은, 이러한 RSS 문서를 통해 주기적으로 관리된다.

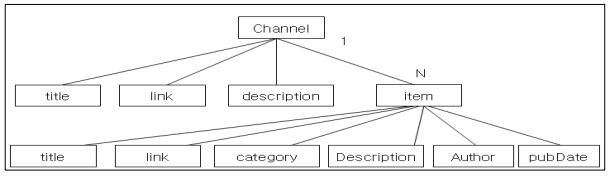

그림 4. RSS 문서의 구조

RSS 채널 관리의 구성은 <그림 5>와 같다. 한 채널에 대한 정보는 RSS 문서 단위로 저장하는 것이 아닌 Item단위로 저장한다. 이는 갱신되는 RSS문서에 포함된 내용이 이전 RSS문서에 비해 전부 새로운 Item으로 구성된 것이 아니기 때문에 저장 공간의 효율성을 위해서이다.

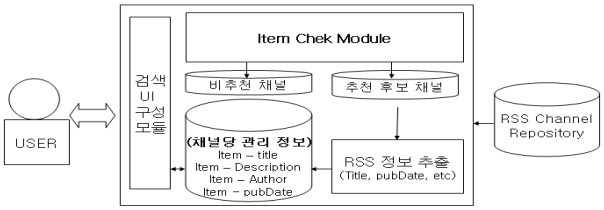

그림 5. RSS 채널 관리 구성도

Item하위의 title과 description정보는 신규 게시물의 제목과 내용부분에 해당된다. 포탈별로 이 부분의 내용에 html 태그가 포함되는 경우가 있으므로, html 태그가 존재하는 경우 정규식을 이용하여 제거한 후 DB에 보관한다. category정보는 블로거가 자신의 블로그에 설정한 카테고리 정보이다. 이는 비슷한 정보에 대해서도 블로거 별로 카테고리 이름을 달리 설정할 수 있으므로 일관성 있는 정보는 아니다. '다음'과 같은 포탈의 경우 이 정보를 RSS에 포함시키지 않고 있다. 따라서 이 정보는 DB에 보관하지 않는다. Author정보는 글의 작성자에 대한 정보이므로, 이를 DB에 저장함으로써 같은 작성자의 다른 게시물도 함께 검색이 가능하도록 지원한다. pubDate는 신규 정보가 올라온 시간을 가리킨다. 이를 DB에 저장함으로써 사용자가 작성 시간에 따른 정보 검색이 가능하도록 한다.

Ⅳ. 시스템의 사용자 인터페이스 구현

RSS채널 탐색 크롤러와 검색 DB를 구현하기 위한 환경은 [표 1]과 같다.

표 1. 구현 환경

구 분	RSS 탐색 크롤러	검색 DB
H/W	Intel Pentium Dothon CPU 1.6 GHz 768 MByte RAM	Intel Celeron CPU 1.6 GHz 512 MByte RAM
S/W	Windows XP pro. JDK 1.5 Informa library	Windows 2000 pro. MS SQL-2000 server

구현된 크롤러는 분당 100개 정도의 RSS 채널의 수집이 가능하며, 주기적으로 접근 시 해당 사이트에서 접근을 막을 수 있으므로, 50개 당 서로 다른 블로그 사이트에 접근한다.

<그림 6>은 메이저 블로그 포탈 사이트의 RSS 채널을 탐색해 나가는 모습을 보여준다. 구현된 RSS 탐색 크롤러는 다음, 엠파스, 하나포스, 코리아닷컴, 메이저 등 총6개의 블로그 사이트의 RSS 채널 검색을 지원한다.

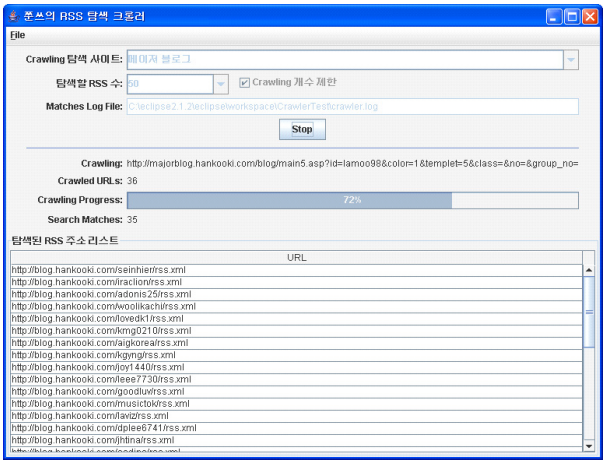

그림 6. RSS 탐색 크롤러

사용자 질의에 대한 구현 화면은 <그림 7>과 같다. 키워드, 작성일, 작성자로 검색 가능하다.

그림 7. 사용자 질의응답 화면

기존의 검색 DB 운용 방법과 비교하면, RSS를 이용한 검색 DB 운용은 다음과 같은 장점을 가진다.

첫째, 웹 문서의 갱신 여부 파악이 쉽다. 기존에는 문서를 가져오기 전 문서가 수정된 날짜를 기준으로 가져오는 방식을 취하였다. 하지만, 이 경우 웹 문서 내 이미지 사이즈 조정 같은, 내용의 변화 없이 레이아웃 변경 시에도 새로운 문서로 인식하는 문제점이 발생할 수 있다. RSS 문서에 포함된 Item내부의 pubDate는 웹 문서의 수정일이 아닌, 신규 내용을 기반으로 생성되는 것이므로, 이러한 문제를 사전에 방지할 수 있다.

둘째, 웹 문서들의 최신 상태 유지가 편리하다. 기존에는 웹 페이지의 수정 부분 파악을 위해 복잡한 분석 작업이 필요하였다. 하지만 RSS의 경우, 수정 부분의 내용추출이 Item하위의Description 앨리먼트로부터 추출 가능하다. 따라서 이를 기반으로 사용자에게 좀 더 정확한 질의응답을 위한 웹 문서의 내용 관리가 가능해 진다.

V. 결 론

RSS와 같은 신디케이션 기술은 사용자에게 신규 정보 여부를 알려줌으로써, 사용자의 불필요한 웹서핑 수고를 덜어준다. 블로그의 증가로 사용할 수 있는 RSS 채널은 많아졌지만, 사용자가 원하는 RSS 채널을 이용하기 위해서는 각 블로그를 직접 돌아다니며 블로그의 내용을 판단하고 RSS 채널 주소를 찾아내야하는 수고를 거쳐야한다.

따라서 본 논문에서는, RSS 탐색 크롤러를 이용한 RSS 채널 자동 탐색과, 사용자의 다양한 질의에 대하여 채널을 추천하는 시스템을 설계 및 구현하였다. 본 시스템은 사용자가 자신의 관심 분야에 대한 정보가 올라오는 블로그를 쉽고, 효율적으로 이용할 수 있도록 RSS 채널을 검색하거나 추천해 준다. 사용자는 추천된 RSS 채널을 RSS Aggregator에 등록함으로써 자신이 원하는 정보를 블로그의 직접 접근 없이 전달받을 수 있다.

향후과제로, RSS 채널의 갱신주기 예측과 사용자의 관심분야를 자동으로 판단하는 연구가 병행되어야 할 것이다.

참고 문헌

[1] PEW INTERNET & AMERICAN LIFE PROJECT, www.pewinternet.org, 2004.

[2] 김성진, "웹 로봇 구현 및 한국 웹 통계 보고", 한국 정보처리 학회 C 10권 4호, 2003.

[3] Soumen Chakrabarti, "Focused crawling: a new approach to topic-specific web resource discovery", In Proceedings of 8th International World Wide Web Conference, 1999.

[4] 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.

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

[6] Haruo Hosoya, "Regular expression pattern matching for XML" Proc. of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming lanuage, pp.67-80. 2001.

[7] BlogKorea, www.blogkorea.org

[8] AllBlog, www.allblog.net

[9] BLOZINE, www.blozine.com