실시간 MP3 파일 검색 엔진을 위한 지원 시스템의 설계와 구현

김우진 (dice@icu.ac.kr)
한국정보통신 대학원 대학교
경영학부 석사과정

최문기 (mkchoi@icu.ac.kr)
한국정보통신 대학원 대학교
경영학부 교수

ABSTRACT

MP3(MPEG 1 layer 3) 파일 형식(file format)은 최근 높은 압축율과 뛰어난 음질 복원 능력으로 주목을 받고 있다. 실제로 MP3 의 압축율은 일반 CD 의 약 50 분의 1 정도이고 음질은 CD 음질을 거의 동일한 수준으로 유지할 수 있다.[1]

한편, 이러한 MP3 의 장점 때문에 web 을 통해 MP3 파일을 찾으려는 수요는 폭발적으로 증가하고 있지만[2] 기존의 검색 엔진들이 가지고 있는 프로세스는 급속하게 update 되고 있는 MP3 컨텐츠에 효과적으로 대응하지 못하고 있는 실정이다. 특히, 기존의 검색 엔진들은 미디어 파일을 위한 검색이 아닌 문자 기반의 검색 기능을 위해 개발되어 MP3 검색에는 부적절하거나, 파일 중심이 아닌 사이트 중심의 링크 변동에 대하여 수동적인 업데이트만을 수행하여 빠른 변화에 능동적으로 대응하기 어려운 경우가 많다.

현재 미디어 파일을 위한 검색 엔진들은 여럿 서비스 중이지만, 텍스트 중심의 탐색 방법을 사용하고, 정기적인 DB update 방법에 관해서도 문자 기반의 검색 엔진과 동일한 방법을 사용하고 있다. 또한, 국내에서는 web 서비스를 위한 미디어 파일 탐색 알고리즘과 지능형 탐색 방법 등에 관한 연구 역시 거의 전무한 상태이다.

본 논문은 MP3 파일 전문 검색을 위한 지능형 프로세스를 설계와 구현 결과에 관한 것으로, 기존의 미디어 검색 엔진들이 가지는 문제점을 지적하고 보다 효율적이고 능동적인 미디어 파일 탐색을 위한 방법을 제시한다. 특히, MP3 파일에 대한 미디어 파일 검증 알고리즘과 verification method 을 제안하고,

[1] 전자 신문 용어 검색 http://www.etnews.co.kr

[2] 조선 일보, 2000.2.8 http://music.chosun.com/site/data/html_dir/2000/02/08/20000208000010.html

이러한 메커니즘에 따라 구현된 지능형 robot 과 spider 등으로 구성된, 신뢰성 있고 지능적인 MP3 검색 엔진 지원 시스템의 설계와 구현 결과 그리고 성능 등을 종합적으로 요약한다.

INTRODUCTION

인터넷을 통한 멀티미디어 파일의 트래픽은 이미 전세계적으로 폭발적인 증가추세를 보이고 있다. 특히 MP3 라 불리는 MEPG 1 layer 3 의 경우에는 국내에서는 여러 검색엔진에서 가장 많이 사용되는 검색어로 자리를 잡았고, 해외의 경우에도 최근 Napster 라는 MP3 음악 파일 교환 프로그램이 급속히 확산되고 있다.

이와 같은 급격한 MP3 미디어 파일에 대한 수요에 비해 국내·외의 여러 검색 엔진들은 아직 멀티미디어 검색에 대한 지원이 충분히 이루어 지지 않고 있으며, 지원하는 경우에도 많은 문제점을 가지고 있다. 국내외의 검색 엔진에서 제공하는 서비스가 가지는 문제점들은 크게 1) 적절한 검색어를 찾기 어렵다는 점과 2) 링크의 변동에 빠르게 대처하지 못한다는 점을 꼽을 수 있는데, 이는 멀티미디어를 지원하기 위한 검색 엔진이 전통적인 문자 기반의 서비스인 Directory Service 나 Search Engine 에서 사용되던 방법들을 사용하기 때문이다.

본 논문에서는 위에서 간략히 제시한 전통적인 검색엔진 서비스에 필요한 기술을 간략히 정리하고 이러한 기술이 멀티미디어 검색을 위해 사용될 때 가지는 문제점을 제시한 후 이러한 문제점을 해결하기 위한 방안으로서의 새로운 검색 엔진 서비스의 작업 프로세스를 제안하고 이를 구현한 결과를 바탕으로 이러한 일련의 작업이 가져오는 효율성에 대해서 기술한다.

또한, 제안된 프로세스에 활용되는 멀티미디어 검증 방법(verification method)을 제시하고 이러한 방식이 기존의 문자 기반의 방식이 가지는 문제점을 해결할 수 있으며, 문자 기반의 방식에 비해 대용량 멀티미디어 파일을 보다 효율적이고 효과적으로 다룰 수 있다는 것을 증명한다.

SEARCH ENGINE TECHNOLOGY

HOW SEARCH ENGINE WORK

SEARCH SERVICE CATEGORIES

인터넷은 무수한 호스트들로 구성되어 있는 네트웍의 네트웍이라고 할 수 있다. 또한, 현재 가장 보편적으로 사용되는 인터넷 응용 서비스는 WWW(World Wide Web)이다.

인터넷이 무한한 정보의 바다라고 불리고는 있지만, 기본적으로 웹 서비스를 통해 제공되고 있는 무수한 컨텐츠들은 호스트를 중심으로 이루어지고 있어 전체의 컨텐츠들 간의 계층성이나, 논리적 연결성이 없다고 할 수 있다.

때문에, 많은 사람들이 검색 서비스를 - 예를 들어 yahoo.com, altavista.com, lycos.com, naver.com - 이용하여 그들이 원하는 정보를 획득하거나 해당되는 정보를 제공하는 사이트를 찾는다. 이러한 검색 서비스들은 사용자의 질의어를 받아 내부적으로 수행되는 일련의 프로세스에 따라 적절한 사이트들 혹은 문서를 추가적인 정보 - 자료의 유사성, 순위, Abstract 등 - 와 함께 제공하는데, 이러한 서비스는 웹 환경에

존재하는 다양한 컨텐츠와 사이트들에 대한 정보를 database 화하여 보유함으로써 가능하다.

검색 서비스들은 크게 2 가지로 나뉘는데, directory service 와 search engine service 가 그것이다.

Directory Service

Directory service 는 Yahoo!가 대표적인데, 웹 사이트 관리자들이 자신들의 사이트를 서비스를 제공하는 회사에 등록하면 디렉토리 서비스 회사는 고유의 주제별 목록에 따라 이를 주제별로 분류하고 정리하여 디렉토리 구조로 변환하고, 이를 바탕으로 서비스를 하게 된다. 따라서 사용자들은 자신이 희망하는 자료를 단계별로 세분화되는 디렉토리를 따라 이동하면서 찾을 있으며, 일반적인 디렉토리 서비스에는 indexing 기능이 있어 검색어를 통해 원하는 자료를 보다 빠르게 찾을 수 있도록 돕는데, 이는 전체 catalog 를 검색하는 기능과 하위 디렉토리만을 검색하는 기능 등으로 나뉜다.

Search Engine

검색 서비스는 text retrieval 기법을 사용하는데, 서비스 회사들이 각기 독립적인 검색어들을 정의하고 robot, web crawler 등을 사용하여 얻어진 웹 문서들을 정의된 검색어를 기준으로 처리하여 이를 데이터베이스에 일정한 구조체로 변환하여 데이터베이스에 보관하게 된다. 이후, 사용자들이 질의어를 던지면 그들의 질의를 정해진 방식으로 데이터베이스를 검색하여 결과를 서비스하게 된다.

TEXT RETRIEVAL

웹은 하이퍼 링크에 기반을 두고 있으며, 웹 문서는 하이퍼 링크로 서로 연결되어 있다. Text retrieval 은 이러한 웹의 특성을 활용하여 웹 문서들을 획득하는 기술로 인트라넷에서나 인터넷에서 robot, web crawler, spider 등의 프로그램을 사용하면 기계적으로 많은 문서를 획득하는 것이 가능하다.

검색 엔진은 사이트의 소유자가 직접 디렉토리 서비스에 등록하는 방식과 달리 검색 엔진 자체의 자료 수집을 통한 인덱스 제작에 기반을 두고 서비스를 수행하기 때문에, 최대한 많은 문서들을 중복 없이 최대한 많이 확보하는 것이 검색 엔진의 성능을 결정하는 가장 기본적인 문제가 될 수 있다.

실제로 text retrieval tool 은 이러한 웹 문서의 수집, 웹 문서의 갱신 여부 확인, 인덱스 제작에 이르는 여러 가지 기능들을 가질 수 있다. 각 기능들에 따라 각기 robot, web crawler, spider 등으로 달리 불리긴 하지만, 큰 범주에서는 거의 같은 의미로 사용된다.

이러한 Text retrieval tool 은 다양한 제품들이 소개되고 있지만, 기본적으로 갖추어야 하는 기능들은 indexing, word searching, updating, HTML hyperlinks handling, crawler depth control, exclusion, tagging, robot exclusion protocol(REP), recovery 등이다.

하지만, 일반적으로 이러한 제품들의 구현된 기능들만으로 하나의 검색 엔진으로 동작하기에 알맞은 시스템을 갖추었다고 할 수 없으며, 검색엔진의 운영 절차나 용도, 데이터 수집 방법들에 알맞도록 적절한 solution 을 구입하여 프로그램을 변형하거나 조작하는 것이 보통이다.

COMPILING INDICES OF DOCUMENTS

문서 획득(text retrieval) 과정을 통해 얻어진 무수한 문서들은 검색엔진에 의해 분류되거나 정리된다. 실제로 검색엔진이 매번 모든 문서를 모두 search 하는 것은 아니다. 문서에 등장하는 핵심적인 용어와 낱말들을 기초로 문서를 분류하고 이를 키워드의 출현 빈도(appearance frequency), 상관 관계에 있는 단어들을 포함하는 문서들에 대한 정보를 데이터베이스화 하여 보관 하게 되는데, 이러한 과정을 보통 indexing 이라 부른다.

이러한 인덱싱 과정에서는 text-mining 기법이 활용되기도 하는데 이는 특히 복수의 검색어를 입력하는 경우 모든 단어를 포함하는 문서를 별도로 처리하여 검색하기 보다 연관성이 높은 단어들을 포함하는 문서들에 대한 정보를 미리 데이터베이스에 추가함으로써 response time 을 줄이고, computing power 를 절약할 수 있는 장점이 있다. 특히 데이터베이스 연산 가운데 join operation 은 가장 비용이 많이 드는 연산이므로, text-mining 기법을 도입하고 빈도가 높은 상관 검색어들을 효과적으로 indexing 하는 것은 검색 엔진의 성능 향상에 큰 도움을 줄 수 있다.

이렇게 보통의 검색 엔진들은 text retrieval 과 indexing 의 방법을 통해 사용자들의 query 에 짧은 시간에 방대한 양의 문서에 대한 검색 결과를 알려 줄 수 있다. 하지만, robot 들이 문서들을 수집하고 이를 인덱싱하는 데에는 엄청나게 많은 시간들이 걸리고, 많은 computing power 를 필요로 한다.

특히 이러한 작업을 수행하는데 많은 시간이 걸린다는 점은 큰 문제점이 될 수 있는데, 이러한 작업들이 수행되는 동안 많은 문서나 링크들이 변경될 수 있다는 것이 그것이다. 심한 경우에는 사이트들이 없어지거나, 기한이 만료되기도 한다.

이러한 문제는 웹 사이트들이 증가할수록 더욱 심각해지고 최근에는 웹이 급격하게 팽창하고 있어 이러한 문제점들은 보다 심각한 문제로 대두되고 있다. 실제로 웹 문서들의 급격한 증가로 인해 무수한 웹 문서들이 검색엔진에 검색 대상에서 누락되는 경우가 발생하고 있으며, 보다 적절히 검색하는데 문제점들을 야기하고 있다.

UPDATING THE INDEX

일반적으로 검색엔진의 데이터 수집작업은 robot 들이 가장 효율적으로 동작할 수 있도록 트래픽이 가장 적은 밤시간에 주로 이루어 지는데, 최신 정보에 대한 여러 중간 조정 작업과 분석 작업을 거쳐야 하기 때문에 수집된 데이터에 대한 update 가 바로 이루어 지는 건 아니다.

대개 주요 검색 엔진들은 정기적으로 목록을 갱신하는 것으로 생각되는 것이 보통이지만, 데이터베이스의 내용에 접근하는 일 자체가 일반적으로는 어려운 일이기 때문에 altavista, hotbot 등의 검색 엔진들은 일주일마다 한 번씩 update 하는 것을 원칙으로 하고 있어도, 보통 이런 작업은 2 주에서 한달 정도의 시간이 소요된다. 따라서 대개 검색엔진에 등재된 데이터들을 한달 정도 된 것인 경우가 많다.

FILTERING TECHNOLOGIES

검색엔진 서비스들의 matching document 의 검색 방법이 주로 문서에 등장하는 특정 단어나

문구에 초점이 맞추어지기 때문에, web designer 들은 보다 높은 우선 순위를 받기 위해 - 즉, 리스트의 상단에 rank 되기 위해 "word stuff"을 사용하기도 했는데, 주로 필터링 기술은 이러한 문서들을 찾아내어 적절하게 대응하기 위한 방안으로 사용되기 시작했다.

LEVELS OF ABSTRACTION

검색 엔진이 질의에 대하여 return 하는 결과물에 대한 추가 정보와 문서의 성격에 대한 정보를 제공하기 위해 문서에 대한 abstraction 을 제공하는데, abstraction 을 생성하는 방식은 각 검색엔진마다 다르다. 주로 이러한 abstraction 생성에는 문서의 초반부를 얼마간 사용하는 것이 보통이다.

또한, abstraction 에 해당되는 문서에 포함되어 있는 링크 정보를 문서의 title 과 함께 제공하는 방법이 사용되기도 하였고, 소위 deep search 라고 불리는 하위 문서를 일정한 - 1 ~ 2 단계 정도의 - depth 까지 검색하여 정확도나 점수를 덧붙이는 방식이 쓰이기도 하였다.

MP3 SEARCH ENGINES PROBLEMS

MP3 형식의 파일이 인기를 모으면서, MP3 형식에 대한 파일을 전문적으로 검색해 주는 검색 엔진들이 많이 등장하고 있다. 기본적으로 이러한 검색 엔진들은 MP3 대용량의 미디어 파일들을 검색의 대상으로 하기 때문에, altavista, lycos 와 같은 문자 기반의 검색 엔진들과 기술적으로 다른 부분이 많다.

기존 검색 엔진들이 MP3 등의 검색 서비스들을 운영하기도 했었는데, 성능면에서 그다지 우수하지 못했기 때문에, 인기가 지속되지는 못했고, 최근에는 MP3 음악과 관련된 warez 혹은 portal site 들이 득세를 하는 양상이다. 문자 기반의 검색엔진이 MP3 검색에 사용되는데 한계를 가지는 문제점들은 크게 다음과 같이 요약할 수 있다.

1. 멀티 미디어 자료를 문자 기반의 text retrieval 과 문서 분석을 통해서 얻는데 한계가 있다.

2. 데이터베이스의 정기적인 update 가 대단히 중요한 문제 인데, 이러한 작업이 문서 중심으로 이루질 경우 시간이 매우 오래 걸리고, 실제적인 데이터인 멀티미디어 파일에 대한 추적이나 update 가 적절하게 수행되기 어렵다.

3. 문자 검색과 달리 효율적인 검색의 방법을 제공하기가 어렵다. - 편리한 검색 방식을 제공하게 될 경우 상당한 추가 비용이 필요하다.

FILE RETRIEVAL & URL MINING

문서 중심으로 서비스를 하는 기존의 문자 기반 검색 엔진들의 경우에는 웹 문서를 획득해 일련의 핵심 단어들을 중심으로 정리하여 이러한 정보에 search algorithm 을 적용하는 방식으로 적절한 문서를 서비스할 수 있지만, 오디오 파일들은 웹 문서들처럼 서로 연결되어 있지 않기 때문에 text retrieval 에 사용되었던 방식을 통해 충분한 데이터를 확보하는데 어려움이 많아진다. 즉, 아직 까지는 이러한 오디오 파일의 위치(URL)를 탐색하는 뛰어난 robot 을 개발하는 문제가 하나의 어려운 점으로 지적될 수 있다.

UPDATING THE DATA

국내외를 막론하고 오디오 파일의 음원에 관한 저작권 문제가 불거지면서 인터넷 상에서 유통되는 오디오 파일들은 대개 warez 사이트나 개인 홈페이지들을 중심으로 제공되는 경우가 많으며 따라서 링크가 변경되거나 파일들이 삭제되는 일이 많다. 그러므로 이렇게 time-sensitive 한 파일들을 대상으로 하는 서비스들은 신속한 자료의 갱신과 데이터의 download 가능 여부를 자주 해 주어야 할 필요성이 발생한다.

이미 지적한 바와 같이 기존의 검색 엔진들은 자료의 갱신을 주(週) 단위로 수행하는 경우가 많은데, 실제로 자료 갱신작업은 많은 시간을 요하는 작업이기 때문에, 근본적인 해결이 어려웠다. 또한 멀티 미디어 자료의 경우에는 비교적 데이터의 용량이 크고 download 에 시간이 많이 소요되기 때문에, download 에 걸리는 시간이나 안정성 등에 대한 추가 정보가 필요하고, 특정한 소프트웨어 프로그램으로 재생 가능한지에 대한 신뢰성을 보장하기 위한 추가적인 시스템이 요구된다.

하지만, 이러한 정보를 제공하기에는 기존의 문서 기반의 검색엔진의 프로세스는 한계를 많이 가진다고 할 수 있다. 현재 검색 엔진들이 MP3 파일 검색 서비스를 하고 있지만, 위에서 제시한 기존의 검색엔진들이 가지는 한계를 가지고 있으며 사용자 중심의 선곡 서비스 등의 부가 서비스를 제공하지 못하고 있다.

SEARCH MATHOD & CUSTOMIZATION

사용자들이 희망하는 유용한 오디오 파일은 대개 음악 파일인 경우가 많기 때문에 이를 분류하기 위해서는 가수 이름이나 연주가 이름, 제목과 음반 등에 대한 제반 정보가 필요한데 이를 획득하는 일들은 text retrieval 과 mining 을 통해 기계적으로 수행하기 어렵다.

또한 아직 범용의 검색 엔진들은 사용자에 대한 기록(profile)을 가지지 않는 것이 보통이다. 따라서 이러한 정보에 따라 각 개인에게 보다 적합한 형태로 가공된 결과를 돌려주도록 지원하는 범용 검색 엔진은 아직 활성화 되지 않았다고 할 수 있다.

하지만, 사용자에 대한 정보와 검색 기록(search query log) 등에 기초한 검색을 지원하는 경우에는 보다 정확한 서비스를 할 수 있다. 특히 오디오 파일과 같이 검색의 범위가 비교적 제한된 경우에는 보다 정확한 검색을 지원할 수 있는 가능성이 있으며, 이전에 기술한 바와 같이 오디오 파일에 대한 음반과 가수, 작곡자, 작사가 등과 같은 부가적인 정보를 바탕으로 하는 경우에는 기호에 따른 정보 제공이 상당히 정확한 수준에 까지 이를 수 있을 것으로 생각된다.

AUDIC.COM - A MP3 LINK DELIVERY SERVICE SYSTEM

다음에서는 위에서 기술한 문자 기반의 검색 엔진이 가지는 문제점을 해결하기 위한 MP3 검색엔진 지원 시스템의 설계와 구현, 성능, 미디어 파일 검증 알고리즘에 대하여 기술한다.

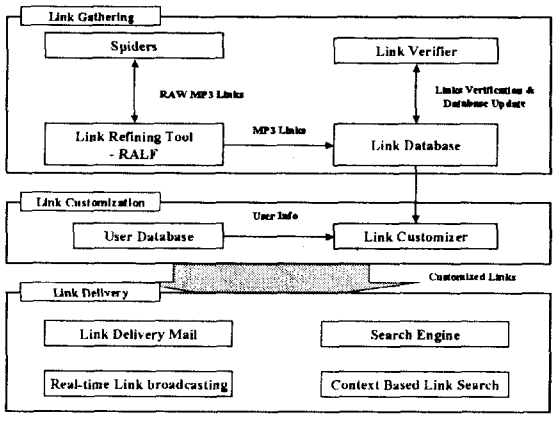
MP3 SEARCH ENGINE SUPPORT SYSTEM

SYSTEM OVERVIEW

다음의 Figure 1.는 전체 시스템의 개념도로, 전체 시스템은 오디오 파일에 대한 정보를 획득하고 가공하여 데이터 베이스화 하는

일련의 과정을 포함하는 Link Gathering 과정, 데이터 베이스의 정보를 사용자의 사용 기록에 따라 보정하여 사용자에게 알맞은 추가 정보를 제공하도록 하는 Link Customization 과정, 이를 사용자에게 다양한 방법으로 전달하는 Link Delivery 과정으로 나뉘어 진다.

Figure 1 Audic.com System Design



LINK GATHERING

Link Gathering 은 일반적인 검색엔진의 text retrieval & indexing 과정에 해당하는 것으로, audio file 간의 상호 참조 링크(hyperlink)가 없기 때문에 발생하는 링크 확보 시의 문제점을 data locality 의 특성을 이용한 spider 와 Link Refining tool(RALF)을 통해 link 를 확보하는 과정과 정기적으로 파일의 존재 여부를 확인하는 MP3 file verifying tool(LV) 등을 통해 진행된다. 또한 이 과정에는 database update 과정을 포함된다.

Locality

Hyperlink 의 개념은 상당히 강력한 것이긴 하지만, 이는 HTML 형식을 따르는 파일에서만 제공되는 것으로 현재로서는 모든 멀티미디어 파일들이 상호 참조 링크를 가지고 있지 않다. 때문에, 범용의 검색 엔진의 경우에는 Link Gathering 과정에서 일반적으로 파일 크기가 작고, 문자 기반으로 이루어진 파일의 분석을 통해 데이터베이스를 구성할 수 있지만, 멀티미디어 검색 엔진의 경우에는 무수한 HTML 파일들을 탐색하더라도 단 하나의 Audio File Link 도 얻지 못할 수 있다는 문제점을 가진다.

하지만, MP3 와 같은 멀티미디어 파일은 주로 특정한 사이트 혹은 HTML 문서에 다수의 링크를 확보하고 있는 경우가 많다. 따라서 일반적인 Spider 에서 사용되는 web 문서 탐색 방식에서 MP3 파일 링크를 가지고 있는 문서의 링크만을 파일로 생성하도록 Spider 프로그램을 변형하면 데이터 링크 확보에 매우 뛰어난 성능을 발휘할 수 있다

또한, 일반적인 Spider 들은 범용의 검색 엔진용으로 설계된 경우가 많으므로, 모든 링크들을 추적하는 방식으로 구현된 것들이 대부분이다. 하지만, MP3 파일의 경우에는 MP3 관련 사이트들 간에 하이퍼 링크가 월등히 많아 이를 MP3 파일을 포함하는 사이트들 간의 링크만을 별도로 관리하면서 추적하면 불필요한 Search 를 줄일 수 있다.

이러한 일련의 과정을 개략적으로 정리하여 기술하면 다음과 같다.

1. 초기 MP3 파일이 존재하는 사이트의 URL 을 찾아 MP3 사이트 URL 버퍼에 추가한다.

2. MP3 사이트 URL 버퍼에서 URL 을 얻어 web 문서를 획득(retrieval)하고 이를 parsing 한다.

3. parsing 한 결과 하여 해당 사이트에 MP3 파일이 존재하는 경우 다른 링크들도 MP3 사이트 URL 버퍼에 추가한다.

4. 과정 3 에서 얻은 MP3 파일의 링크는 MP3 파일 링크의 버퍼에 이를 획득한 문서의 URL 을 기록하고 과정 2 로 돌아 간다.

이러한 일련의 과정을 반복하면 MP3 사이트의 URL 을 추가적으로 확보할 수 있을 뿐만 아니라, MP3 파일의 URL 을 포함하는 문서들의 URL 을 효율적으로 얻을 수 있다.

이러한 방법은 모든 웹 문서를 탐색하고 이를 분석하여 얻어진 링크들이기 때문에 웹 상에서 제공되는 모든 MP3 파일들을 탐색하지는 않지만, 일반적으로 MP3 등의 오디오 파일들은 몇몇 제한된 portal site 나 개인 homepage 에만 있기 때문에 - Link Locality 가 있기 때문에 이러한 알고리즘을 이용하면 robot 의 작업량을 최소화하면서 많은 링크를 효과적으로 확보할 수 있다.

compiling indices of files through guessing – RALF(real-time audio location finder)

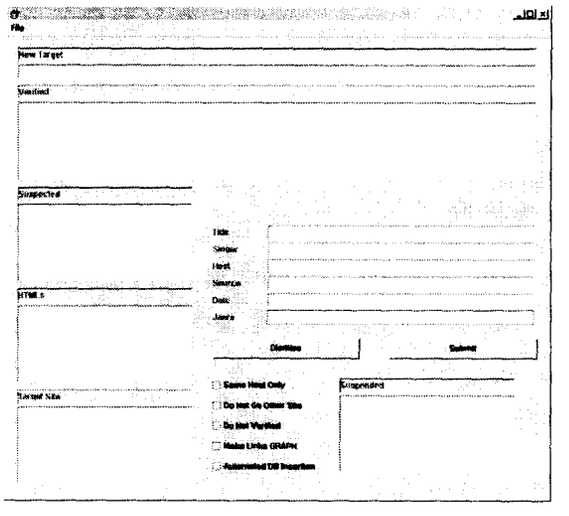
Audic.com 서비스를 위해서 전용 browsing tool 을 개발하였는데 RALF(Real-time Audio Location Finder)라고 명명되었고, 현재 version 0.5 가 개발되어 반 자동화(semi-automation) 된 링크 인덱싱 작업을 지원하고 있다.

음악 파일의 정보 검색 작업은 매우 정형화된 정보 즉, 가수, 작곡자, 앨범 이름, 제목 등의 정보들을 바탕으로 검색이 가능하다. 하지만, 웹 문서만을 분석하여 이와 같은 정보를 정확하게 획득하는 것은 매우 어려우며, 추가적인 음악의 정보를 데이터 베이스에 확보하고, 이를 상호 비교하는 과정이 필요하다.[11]

RALF 는 파일의 링크 전후에 존재하는 여러 정보를 일련의 Guessing Rule 을 적용하여 MP3 파일의 링크를 제거하거나, 곡의 제목, 가수의 이름 등에 대한 추정값을 프로그램 사용자에게 제공한다. 프로그램의 사용자는 이러한 정보의 획득과정에서 일어날 수 있는 오류를 정정하고, 추가적인 정보를 제공하여 데이터베이스에 보관하게 된다.

다음의 그림은 RLAF 의 실행 화면이다.

Figure 2 RALF 실행 화면



이러한 방식으로 정형화되고 정제된 정보는 범용의 기존 검색 엔진의 경우에 단순히 일정한 key word 를 포함하는 문서의 일부를 정리하여 이를 서비스하고 있기 때문에 가질 수 없는 보다 강력하고 정확한 검색의 결과를 제공할 수 있다..

추후 많은 곡들에 대한 정보를 database 화하고 이를 응용하면 보다 나은 guessing 능력을 가질 것으로 생각되며, 현재 Java 기반 시스템으로

가지는 제약을 극복하게 되면 좀더 나은 수준의 자동화를 달성할 수 있을 것으로 예상된다.

Verifying & updating the links – LV(Link Verifier)

Link 의 update 와 verifying - 파일의 존재하여 다운로드를 받을 수 있는지 여부를 확인하고, 파일 형식에 알맞은지 검증하는 과정 - 을 위해 개발된 프로그램은 Link Verifier 라 명명되었는데, 이 프로그램은 데이터 베이스나 파일에 들어 있는 링크들의 정보를 확인하여 URL 이 지시하는 파일이 존재 여부와 다운로드 속도, 파일 형식 등을 확인한다.

또한 최근 파일 verification 시간과 날짜를 time stamp 로 기록하여 함께 제공함으로써 사용자의 결정을 위해 추가적인 정보를 제공할 수 있도록 한다. Verification 에 실패한 파일은 기록되었다가 사용자들에게 제공되지 않도록 별도로 관리되며, 일정한 기간 마다 다시 validation 과정을 거쳐 검증되지 않으면 이를 제거한다.

LV 는 파일의 검증의 효율화를 위해 미디어 파일의 일부만을 확인하는 방식을 사용하는데 이때, MP3 파일 양식의 Header 를 확인하고 이의 빈도와 프레임의 길이 등의 정보를 바탕으로 Guessing Rule 을 적용함으로써 보다 정확한 확인 작업을 수행하고 일련의 과정을 자동화 한다.

LV 를 통한 검증 절차는 사용자들에게 time-sensitive 한 파일에 대한 사용 가능한 정보만을 제공할 수 있도록 지원함으로써 검색 엔진에 대한 신뢰를 가질 수 있도록 한다.

또한, 미디어 파일만을 대상으로 해당 파일의 일부만을 통해 검증할 수 있도록 효율화된 Guessing Rule 을 적용함으로써 범용의 검색 엔진에서 필요했던 오랜 데이터베이스 update 시간을 줄임으로써 하루 한 번 이상의 데이터 베이스 update 가 가능하다.

SUMMAY & CONCLUSION

인터넷의 컨텐츠와 파일들의 연결 관계가 수평적이기 때문에 웹 서비스가 활성화 되면서 이러한 컨텐츠와 사이트들을 논리적으로 연결한 필요성이 생겨났고, 이러한 연결성의 제공은 디렉토리 서비스에서 검색 엔진 서비스로까지 이어져 모든 인터넷 사용자가 이용하는 중요한 서비스로 자라나게 되었다.

디렉토리 서비스는 관리자의 등록 요청에 의해 등재되는데, 세분화된 분류체계에 알맞도록 각 사이트를 나누어 사용자들에게 제공하는 서비스이고, 검색 엔진 서비스는 text retrieval tool 을 이용해 얻어진 방대한 데이터를 고유한 방식으로 분류하여 데이터베이스화하고 정해진 검색 방식에 따라 사용자들의 query 를 처리하여 결과를 제공하는 방식으로 이루어는 서비스이다.

하지만, 검색 엔진 서비스는 문서 기반의 서비스에 알맞은 형태의 서비스로 이를 멀티미디어 검색 서비스를 제공하는 데에는 한계가 있다. 실제로 이러한 서비스가 가지는 문제점은 크게 검색의 대상인 멀티미디어 파일들 간의 상호 연결성(hyper link)이 없고 멀티미디어 파일들의 링크가 집중화 되어 있다는 점과 멀티미디어 파일에 대한 부가 정보는 문서 검색으로 얻어질 수 없다는 점으로 요약된다.

여기에서는 실제로 준비 중에 있는 MP3 검색 엔진인 audic.com 을 기초로 기존의 검색 엔진 기술이 어떻게 멀티미디어 검색 엔진 개발에

응용되고 문서 중심의 검색엔진이 멀티미디어 검색 기능을 제공하고자 할 때 발생할 수 있는 문제점들을 어떤 방법으로 해결하였는지에 대해 간략하게 소개하였다.

특히 멀티 미디어 파일의 집중화 현상을 다루기 위해 적용되었던 Robot 의 추적 알고리즘 개선 작업과 링크 gathering 을 위해 개발된 RALF 또한 링크의 업데이트와 verification 을 위해 개발된 LV 에 사용된 지능형 알고리즘을 개략적으로 기술하였으며, 추후 이러한 일련의 과정을 통해 획득 되어진 데이터를 효율적으로 사용자들에게 제공하기 위한 방법도 간략히 소개하였다.

Robot 의 MP3 파일 위치 탐색에 알고리즘이 일반 검색엔진의 데이터 gathering 방법에 비해 MP3 관련 데이터의 획득에 있어서 탐색 시간과 트래픽을 월등하게 줄이면서 효율적으로 동작할 수 있도록 사용되었으며, 이렇게 발견된 문서들은 RALF 라는 browsing tool 을 통해 정재되어 데이터 베이스에 저장되도록 설계되었다. RALF 의 도움으로 전체 시스템은 정확한 정보를 바탕으로 사용자들에게 다양한 검색 방법을 지원할 수 있으며, 이후 LV 에 의해 위한 관리함으로써 데이터의 신뢰성을 높힐 수 있었다.

이미 간략히 소개한 Customization 과 보다 능동적인 검색 서비스로 발전하기 위한 부가 기능들에 관한 연구와 보다 다양한 미디어 파일들에 응용하는 문제들이 향후의 과제로 남아 있다.

REFERENCE

COMPUTER TECHNOLOGY RESEARCH CORP(1999), *SEARCH ENGINE TECHNOLOGY FOR THE WORLD WIDE WEB AND INTRANET*

전자 신문, 용어 검색 검색 *HTTP://WWW.ETNEWS.CO.KR*

조선 일보, 2000.2.8 *HTTP://MUSIC.CHOSUN.COM/SITE/DATA/HTML_DIR/2000/02/08/20000208000010.HTML*

AUDIC.COM CORP(2000), *AUDIC.COM 기술 문서*

[i] 보다 세부적인 알고리즘은 Audic.com 검색 엔진과 관련된 내용으로 대외비이므로 개략적인 알고리즘만을 제시

[ii] 현재 이러한 기능을 구현하기 위해 데이터 베이스와 이에 적용하기 위한 알고리즘을 연구하고 있다.