AROOOGA: An Audio Search Engine for the World Wide Web

Ian Knopke
Digital Distributed Music Library Laboratory
Faculty of Music, McGill University
ian.knopke@mail.mcgill.ca

Abstract

Existing search engines use web crawlers to gather web pages. The extracted information is used to build indexes, which are later used to answer user queries. This approach is useful for general queries, but ignores the special properties of sound files, making it difficult to accurately locate specific sound files on the web. AROOOGA, or the ***A****rticulated* ***R****esource for* ***O****bsequious* ***O****pinionated* ***O****bservations into* ***G****athered* ***A****udio, is a web crawling system designed specifically to find and analyze audio resources on the web. The AROOOGA web crawler uses both audio information and the associated web pages to produce higher-quality search indexes for music information retrieval. Information about sound files on the web is discussed, and some preliminary search results are included.*

1 Introduction

The World Wide Web is decentralized by design; there is no inherent master index. This is not a problem for casual browsing, but introduces difficulties for users attempting to locate specific information relating to topics and interests. This need is usually addressed through the use of a search engine, an online interface for matching user queries against a previously constructed index of web content. The formation and makeup of this index is vital to the quality of the results returned by the search engine, as well as the user experience.

One method for creating such an index is for human agents to manually index web content using descriptions, keywords or classification categories, an approach that has been used in both the Yahoo and Lycos search engines. However, this approach can be problematic. While acceptable for smaller data set, the current size and growth of the web makes it difficult for humans to keep up with the sheer volume of material. also, it has been shown that humans may vary greatly in their descriptions of material or may be inconsistent, introducing classification errors into the index.

Alternatively, most search engines such as Google or Altavista build their indexes using automatic analysis systems know as web crawlers, spiders or robots. These systems work by downloading web pages, performing an analysis, and following the extracted hyperlinks to other pages, and then repeating this process indefinitely. The process can be almost entirely automated, and while not without some error, the analysis process is extremely consistent.

Most existing web search engines are designed to answer general-purpose textual queries. This works by converting each downloaded page into a series of text tokens, and indexing those tokens. While useful for general-purpose queries, this approach is less effective for audio materials. Sound files that are linked to from web pages are treated exactly the same as any other text attribute, ignoring the special properties of these materials. Attempting to locate sound files using standard search engines results in pages that mention the query terms; there is no guarantee that that page will actually contain a link to a sound file, even if the file is named explicitly. This makes searching for music on the web a frustrating experience, and greatly reduces the effectiveness of the web as a music distribution system.

This paper describes the web crawler associated with a search engine that has been designed specifically to crawl the web for audio files, perform both text-based and audio analyses, and use the resulting information to build better query systems for the location of audio on the web. By using both audio analyses and the associated text-based information from the linking page, at the crawler level, it is possible to build more accurate systems than could be produced using either type of information in isolation.

2 Related Work

The first web crawler was Matthew Gray's Web Worm, (Gray 1995), designed to measure the size of the web at that time. Search engine technology has flourished since then, and some simple crawler / download systems are now freely available (Anonymous 2004). While suitable for small crawls, most of these are systems are not designed for gathering and analyzing large numbers of pages over extended periods of time. High capacity crawlers are found in search systems such as Google, AltaVista, and Yahoo. Unfortunately, these ventures are proprietary in nature, making it difficult to get detailed information about the technology involved. In general, these systems are extremely robust and get increased crawling speeds by downloading multiple pages in parallel.

A description of the original Google search engine, as it existed as a Stanford University research project is available (Brin and Page 1998). At this time, Google used a group of at least eight computers to download multiple pages from the Stanford network in parallel. Mercator (Heydon and Najork 1999) is a high-capacity web crawler from Compaq Systems Research Center which can be easily configured to handle different types of protocols, such as HTTP, FTP, or GOPHER. Mercator formed the basis of the original Altavista engine, and has been made available for select research projects. Another recent multithreaded Java-based crawler uses multiple machines connected using the Network File System (NFS) to transfer data between nodes (Shkapenyuk and Suel 2002).

Other crawler designs have been described in less detail (Burke 2001; Boldi, Codenouti, Santini, and Vigna 2002). Crawling technology for multimedia is still in its infancy, with most development in this area applied to still images (Rowe 2002; Sclaroff, Taycher, and La-Cascia 1997). The CUIDADO project is an audio and music content analysis system which will eventually incorporate a web crawler (Vinet, Herrera, and Pachet 2002).

3 AROOOGA

AROOOGA is a web crawling system designed specifically to find and analyze audio resources on the web. It is a multiprocessed system that achieves high crawling speeds by downloading multiple pages in parallel. Much of the system is written using the PERL programming language. AROOOGA works by downloading web pages and analyzing them for links to both other web pages and to audio or music materials. The system consists of two types of components, a set of retrievers and a coordinating crawl manager. A simplified diagram of the system is given in Figure 1.

Web pages are downloaded by *retrievers*. Usually multiple retrievers are active at any one time. A retriever has a simple lifecycle: it requests a new URL from the crawl manager, downloads the materials at that URL, applies the appropriate analysis routine or routines, and transmits the extracted information back to the crawl manager for storage. Retrievers are capable of retrieving both web pages and audio materials. Appropriate analysis routines are applied based on the type of material requested.

The *crawl manager* coordinates the fleet of retrievers, by parceling out new links as requested, and handling all storage of data returned by the retrievers. This has two advantages. First, it abstracts all data storage mechanisms within a single program. Secondly, it makes it possible to manipulate the order in which URLs are sent to the retrievers, and hence the order in which pages are crawled. This is important for adaptively prioritizing which web pages are searched first and focusing the entire crawler on certain types of materials.

The speed at which AROOOGA gathers materials is governed by the number of retrievers active in the system. Upon initialization, the first job of a retriever is to make contact with the crawl manager, claiming a new socket for all future transactions. New retrievers can be initialized at any point during a crawl, and there is no requirement that they must be active on the same machine as the crawl manager. All that is required is TCP socket connection, and a sufficient level of system resources.

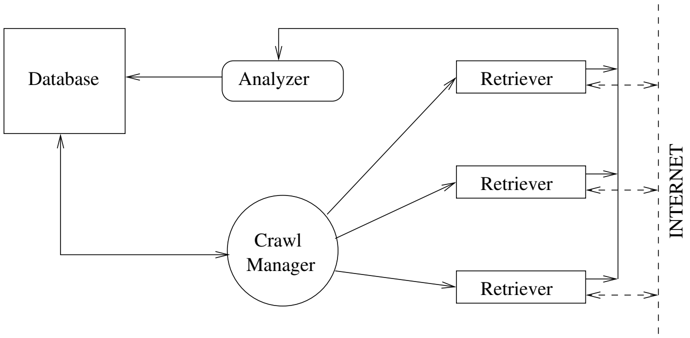

Figure 1: Simplified diagram of the AROOOGA architecture

4 Music on the Web

The goal of this system is to create better indexes of audio information on the web than are available from full-text search engines, by using all available information about the file, including textual and audio information. Textual information is available from the web page that provides the link, as well as keywording within the file. Audio information is analyzed directly from the file. By using all available information, the quality of the index can be improved immensely. This also makes it possible to submit more advanced queries. For instance, queries for “reggae” + “stereo” + ”faster than 100 BPM” could be undertaken, a query which would produce very poor results using current systems.

Information about music or audio files on the web can be divided into three basic categories.

4.1 External Metadata

This is information which is found on a web page that refers to or is associated with a sound or music file. For a sound file to be accessible on the web, there must be a link to that file on a web page. This link also forms an implicit bond between the sound file and the *content* of the web page. The similarity in content between two linked pages is referred to as topical locality (Davison 2000) and is one of the aspects of the web that make it so attractive to most users.

Potentially, any of the information on a page is available for classification. In practice, it is necessary to limit analysis to a few specific types which are easy to extract. Hyperlinks consist of two parts, a visible comment which the user clicks on, and an unseen URL reference to the associated sound file. However, it has been shown (Chakrabarti, Dom, Gibson, Kleinberg, Raghavan, and Rajagopalan 1998) that the information immediately around the link is often related to the link itself, and by extension, the sound file. By defining a word boundary on either side of the link it is possible to also capture this information, for instance by extracting a limited number of word tokens on either side of the link.

This type of information is the easiest to obtain, as it is necessary to analyze the web page as part of the normal crawling process. It is considered to be external to the sound file, because it does not require the sound file to be downloaded.

4.2 Internal Metadata

This refers to information about the sound file which is included within the body of the file. This type of metadata is referred to in the MPEG standard as "ancillary information". Two types of data are available here. The first is technical information about the file and the recorded audio, such as the length of the file, the sampling rate, bit rate, type of compression (if used), or whether it is a monaural or stereo recording. This information is required by playback devices to play the sound file properly. Most sound file formats also make allowances for storing keywords or descriptions. This may include the composer or author's name, genre of the piece, or even full textual descriptions.

Information in this category is potentially more accurate than external metadata, because it is usually attached to the file at the time of creation, and requires specialized software to alter.

4.3 Sound Representation

Every audio file has some way to encode music or sound within it. This category usually refers to recorded digital audio, but could also refer to MIDI or other symbolic music encodings; from the point of view of a retriever, it is simply a matter of calling the appropriate procedure. AROOOGA's existing capabilities are designed to be extended for specialized types of analysis through the use of external bodies of code. This is to accommodate both the large body of analysis research which exists, as well as future developments in audio analysis. Additional capabilities can be easily incorporated into the system by means of a template wrapper system. Currently, AROOOGA employs the MARSYAS system (Tzanetakis and Cook 2000) for the extraction of genre and tempo directly from the recorded audio.

General Crawl Characteristics	
Links collected	16355437
Attempted downloads	253828
Successful downloads	241894
Unsuccessful downloads (404 errors, etc.)	11934
Web Page Links	
Total links to web pages	15983878
Average links per page	66.08
Minimum links	0
Maximum links	12788
Sound File Links	
Total web pages with links to sound files	1651
Total sound files linked to	13012
Average sound links per page	0.054
Average sound links on sf pages	7.88
Minimum links (sound file pages)	1
Maximum links (sound file pages)	205

Table 1: Web Audio Crawl Statistics

Type	Number	Percent	Avg Size(Kb)
MPEG	10991	84.47	1546
WAVE	1501	11.54	732
AIFF	490	3.77	1905
OTHER	30	0.22	422
TOTAL	13012	100	1407

Table 2: Sound File Types

5 Preliminary Results

A recent breadth-first crawl was conducted of approximately a quarter of a million pages. Some statistics of the crawl are presented in Table 1. Audio data on the web is quite sparse, as compared to the number of web pages. When analyzed, only 1651 pages were found to contain links to audio files. However, the accumulated number of pages produced links to 13012 sound files. This demonstrates that most sound files tend to be clumped around a smaller number of central pages. Of the types encountered, the majority were found to be MP3 files, with WAVE file types a distant second (Figure 2).

Each web page will likely have multiple links to other pages, or to sound files. The growth rate of URLs will be much greater than the number of web pages, which in turn exceeds the growth of sound files discovered on the web. This is presented in Figure 2.

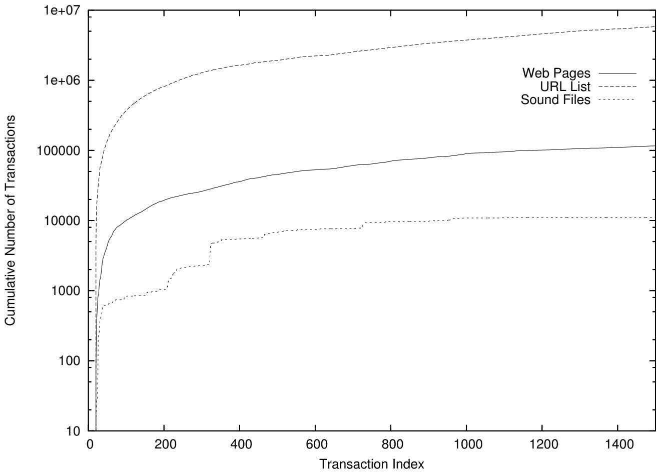

Figure 2: Growth rates of URLs, Web pages parsed, and sound files discovered

6 Conclusions

Unlike specialized delivery systems such as Kazaa and iTunes, there is a wealth of audio materials currently available on the World Wide Web. Much of this is inaccessible due to poor indexing in traditional search engines that concentrate on text-based information and do not make special provisions for audio and music files. This paper presents an overview of a system for web crawling sound and music files, to be used in creating more-specific indexes for the AROOOGA search engine. A classification hierarchy of available information for web-based sound files is presented. Preliminary statistical results show that audio files on the web are typically found on less than 1% of pages crawled. The majority of those files tend to be MPEG types (84.47%).

7 Future Work

To date, most testing crawls have been less than a million pages. This is primarily a limitation of the computing power being used. Much larger crawls, in the range of hundreds of pages, are being planned. Only the MARSYAS audio analysis toolkit is being used. Support for other freely available tools will likely be incorporated into the system. The search engine is restricted to a small number of people on our local network. A high-capacity hosting service is required for use by the general public.

References

Anonymous (2004). Gnu wget. `http://www.gnu.org/software/wget/wget.html`.

Boldi, P., B. Codenouti, M. Santini, and S. Vigna (2002). Ubicrawler: A scalable fully distributed web crawler. In *Proceedings of the Eighth Australian World Wide Web Conference*, pp. n.p.

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. *Proceedings of the 7th International World Wide Web Conference*, 107–17.

Burke, R. (2001). Salticus: guided crawling for personal digital libraries. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 88–89.

Chakrabarti, S., B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan (1998). Automatic resource list compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*. n.p.

Davison, B. D. (2000). Topical locality in the web. In *Proceedings of the 23rd Annual Conference on Research and Development in Information Retrieval*, pp. 272–79.

Gray, M. (1995). Measuring the growth of the web: 1993 to 1995. `http://www.mit.edu/people/mkgray/growth/`.

Heydon, A. and M. Najork (1999). Mercator: A scalable, extensible web crawler. *World Wide Web 2*(4), 219–29.

Rowe, N. (2002). Marie-4: a high-recall, self-improving web crawler that finds images using captions. *Intelligent Systems, IEEE 17*(4), 8–14.

Sclaroff, S., L. Taycher, and M. La-Cascia (1997). Imagerover: a content-based image browser for the world wide web. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pp. 2–9.

Shkapenyuk, V. and T. Suel (2002). Design and implementation of a high-performance distributed web crawler. In *Proceedings of the 18th International Conference on Data Engineering*, pp. 249–54.

Tzanetakis, G. and P. Cook (2000). MARSYAS: A framework for audio analysis. *Organized Sound 4*(3), 169–75.

Vinet, H., P. Herrera, and F. Pachet (2002). The Cuidado project: New applications based on audio and music content description. In *Proceedings of the International Computer Music Conference*, pp. 450–454. ICMA.