An Audio Search Engine for the Web

Ian Knopke

Digital Distributed Music Library Laboratory, McGill

How big is the World Wide Web?

No one really knows. However, there are some estimates:

`1993` A few thousand pages (Shkapenyuk 2002)

`1997` About 3 billion pages (Chakrabarti 1998)

`2000` More than 7 Billion Web Pages (Murray 2000)

`2003` ?

This is a growth of over *200 million percent*! ... or about an order of magnitude per year.

This is an enormous amount of information!

What to do about it

The reason it is difficult to determine the size of the web is that it is *decentralized*. No single node is more important than any other.
This also means that there is no master index of the World Wide Web, which also makes it difficult to find things.
There are a couple of choices.

- You can browse around and hope to run into interesting things.
- You can use a type of software called a *search engine*.

Tell me more about Search Engines

Some examples of common search engines are *Google* or *Altavista*.
How does it work?

- Through a web-based user interface, queries are submitted to a database.
- The database looks through its information and selects results that match the query.

Filling the database

How does the database get filled in the first place?

- One way is to get a lot of people to look at web pages, classify them, and enter them by hand into the database. This is the approach used by Yahoo. It is expensive, time-consuming, and prone to error
- Another approach is to use a *web crawler*

OK, so what is a web crawler?

A Web crawler is a program which automatically indexes pages from the Internet, following a simple plan:

1. A web page is downloaded from the address on a list of web links.
2. The text of the page is analyzed; the analysis is stored in the database for use in user queries.
3. The page is also parsed for any links to other web pages, which are placed on the list of links.
4. The web crawler then downloads one of the linked pages from its list, analyzes it and so on.

The crawler stops when it runs out of web pages or is given new instructions.

Adding a few improvements

A more realistic web crawler is compartmentalized into several specialized sections.

- Usually there will be multiple *downloader* modules, each of which can download a single web page at the same time.
- To coordinate the downloaders and avoid overlap, multiple downloaders are usually controlled by a *crawl manager*.

A model of a Web Crawler

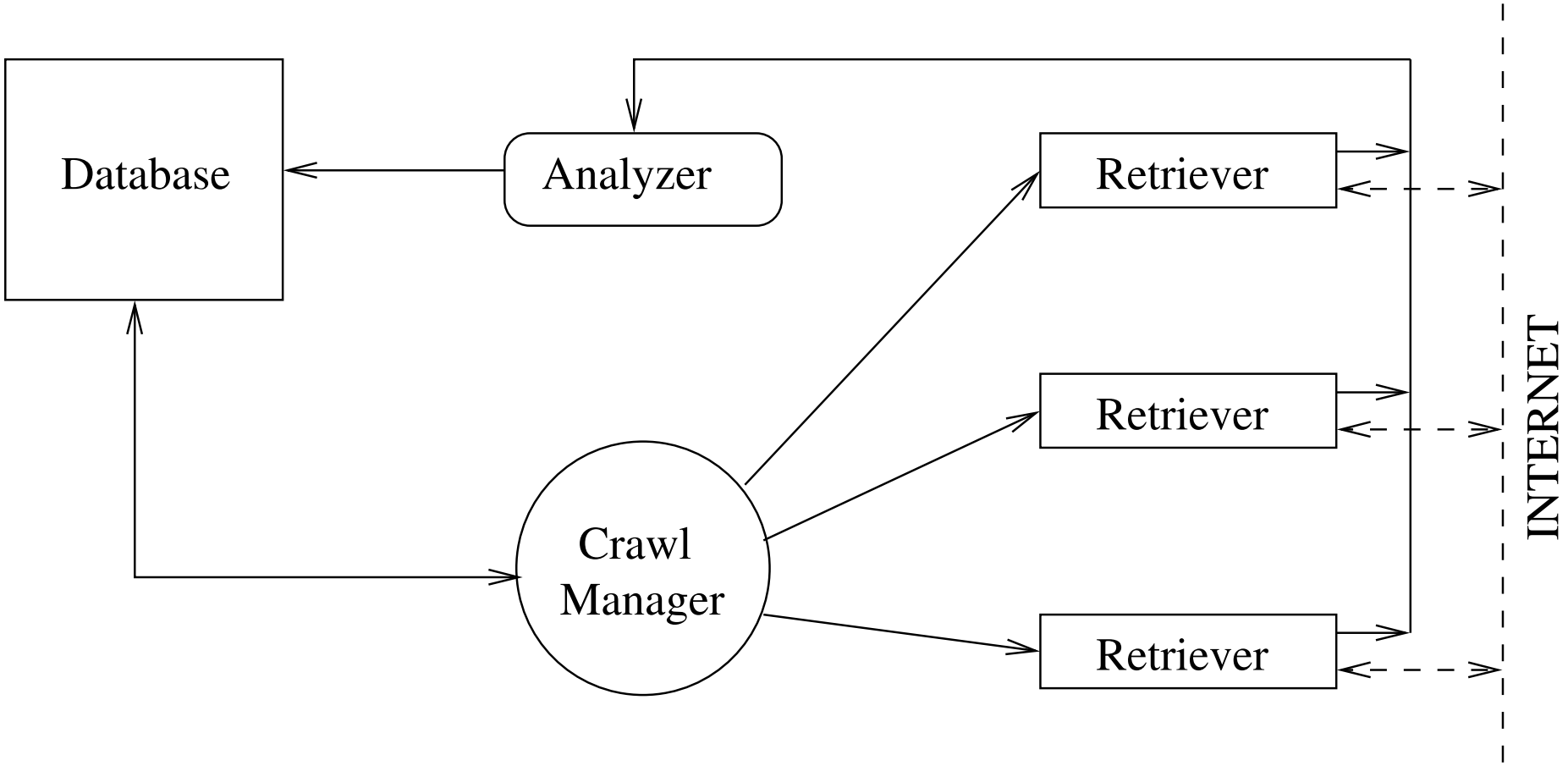

Figure 1: Example web crawler

Importance of a good web crawler

Even though a web crawler is the least visible part of any search engine, In many ways it is the most important part.
In general, the efficiency and design of the web crawler determine two characteristics of the database:

- The total amount of data, or *breadth* of the database.
- How up to date the database is, known as the *freshness*.

There are some very efficient models of web crawlers, but research in this area is largely directed towards text. Some research is also being done into images and video.

No one is working on sound in any serious way!

Sound on the Internet

Audio is stored on computers as *sound files*. There are many different types of sound files, and examples of most types such as wave (.WAV), aiff (.AIFF), midi (.MID) and mpeg (.MP3) can be found on the Web.

What kind of data can we gather from a sound file on the Web?

I divide this into three categories:

External References (references to the sound file)

Internal Metadata (data contained within the sound file)

Encoded Audio (the actual recorded audio)

External References(I)

This is the information which is available within the web page. Note that the sound file is not actually part of a web page. Instead, the web page contains a code for the location or a *reference* to the sound file.

More specifically, this includes:

- The location of the sound file
- The name of the sound file
- The lexical information encoded in the reference

Additionally, it may also be possible to use some of the other text around the link (contextual analysis), but this is difficult to analyze.

External References(II)

```
<A HREF=
"http://www.music.mcgill.ca/~knopke/
    clown.aiff">
    GWB
</A>
```

Location: `http://www.music.mcgill.ca/~knop`

File Name: clown.aiff

Link Name: GWB

This is all of the audio analysis most search engines do. In other words, they treat sound files as if they were any other piece of text.

Internal Metadata

This refers to information that is included within the body of the sound file but which is not actual sound information.

Some examples:

Text-based data composer's name, time signature, or a genre categorization of the piece.

Numeric data length, number of recorded channels or compression type.

The upcoming MPEG-7 standard will make extensive use of this type of information.

Encoded Audio(I)

This is the actual sound data that is heard through speakers when the sound file is played back. This information can be found in several forms.

- The audio information may be a straight PCM encoding scheme, as in some versions of the WAVE and AIFF file formats
- Compression could also be applied, as in MPEG encoding
- Information may also be encoded as a representation of the sounds, as in MIDI where audio is represented as a series of pitches and durations.

Encoded Audio(II)

There is a large body of existing research on extracting features from audio information, especially with regards to speech signals. Much of this research is applicable to a search engine. However:

Time is a factor. It is important that the crawler not spend too much time on any one file. Because of this, simpler procedures that can be done quickly are preferred over complicated procedures that take exceptional amounts of computing power.

Simpler sets of features can often be combined for greater analysis accuracy.

Problems with Traditional Crawlers

Traditional crawlers and search engines are focused on text and text-based analysis methods. This is because they have to index “everything”.

When analyzing sound files, they have tended to apply these textual methods to, with less-than-perfect results, i.e. only using the titles. This ignores much of the information that is specific to sound files and transmission.

In a search engine that is focused on audio files, we can concentrate on using all of the information that is available.

Improvements

How can this information be used to improve the traditional search engine model?

- Additional information can be used to improve the quality of the results returned from a query. This leads to improvements in accuracy and relevance, as well as types of searches not available under other models (stereo file + composer name + [BPM > 120])
- This information can also be fed back to the crawl manager of the web crawler and used to direct the crawler towards areas of the web which are more likely to produce relevant pages.

Software Metrics

How do we know if this model of a search engine works better than traditional models?

There are two ways:

- We can examine the quality of the results returned. A search engine is “better” if it returns the best links first in its list of results.
- Speed of execution, crawling, and analysis may also be useful to evaluate.

Current State of the Project

- Most of the existing work has been done on the Crawler.
- A single process crawler was implemented in Perl, which was capable of downloading up to 25 pages a second.
- Recently, a multi-processed crawler has been implemented which can run on multiple computers in a distributed fashion.
- External metadata is being parsed, and some internal metadata procedures have been implemented
- Analysis of various audio analysis procedures has been done in Matlab but have not been linked to the crawler as yet

Questions?

Any questions?