A MUSICAL WEB MINING AND AUDIO FEATURE EXTRACTION EXTENSION TO THE GREENSTONE DIGITAL LIBRARY SOFTWARE

Cory McKay
Marianopolis College and CIRMMT
Montréal, Canada
cory.mckay@mail.mcgill.ca

David Bainbridge
University of Waikato
Hamilton, New Zealand
davidb@cs.waikato.ac.nz

ABSTRACT

This paper describes updates to the Greenstone open source digital library software that significantly expand its functionality with respect to music. The first of the two major improvements now allows Greenstone to extract and store classification-oriented features from audio files using a newly updated version of the jAudio software. The second major improvement involves the implementation and integration of the new jSongMiner software, which provides Greenstone with a framework for automatically identifying audio recordings using audio fingerprinting and then extracting extensive metadata about them from a variety of resources available on the Internet. Several illustrative use cases and case studies are discussed.

1. INTRODUCTION

Users of modern digital music collections benefit from many advantages relative to users of even a decade ago. Amongst the greatest of these advantages is cheap and convenient access to diverse and rich on-line sources of musical data and metadata. Of particular convenience to researchers and programmers, many on-line sources provide access to their data through convenient web service APIs. Such resources include The Echo Nest, Last.FM, MusicBrainz, Amazon, Yahoo! and many others.

It is also possible to extract features directly from both audio and symbolic musical representations. The resulting feature values can then simply be stored directly as part of digital music collections. Alternatively, these features can be processed using data mining techniques in order to arrive at additional metadata, such as class labels or links to other musical entities.

It is necessary to overcome certain important challenges in order to effectively take advantage of the plentiful data and metadata that is available, however. One must find efficient and effective ways of automatically accessing and integrating information about a given music collection from the diverse and often inconsistent on-line resources; one must ensure that proper identifiers are used to uniquely refer to the individual entities about which information is accessed (e.g. recordings, albums, musicians, etc.), even when the different resources from which data is extracted may identify entities in entirely different ways; one must filter out noisy or inaccurate information, which can be a significant problem when dealing with much of the musical data that is available on-line; one must structure acquired data so that it can be queried and otherwise accessed in ways that are consistent and meaningful; and one must make the data accessible to users in ways that are convenient to them in a variety of use cases.

This paper presents an upgrade to the well-established and open-source Greenstone Digital Library software [10] that is intended to address these issues. This upgrade dramatically expands Greenstone's ability to collect musical information and make it conveniently available to users. Part of this upgrade includes the integration of parts of the jMIR [8] music information retrieval software into Greenstone, specifically jAudio [7,8], which allows content-based features to be extracted from audio recordings.

The second major component of the Greenstone upgrade is the creation and integration of the new jSongMiner software, which provides a framework for automatically acquiring and structuring many types of metadata from diverse sources of information about music, including both on-line resources and metadata embedded in files. This software is highly configurable, in order to meet the needs of a wide variety of different user types. It is also specifically designed to be easily extensible so that different kinds of information can be extracted from different data sources as they become available.

So, given a set of musical recordings of interest, users can now have Greenstone automatically identify unknown

recordings—or verify the identity of labelled recordings—using audio fingerprinting, extract a wide variety of metadata from different on-line sources related to each recording, extract content-based features from each recording and extract any metadata embedded in the tags of each recording. All of this data is then automatically integrated, structured and saved.

Users may then take advantage of Greenstone's established interface to organize, browse or search the newly-built music collection. They may also use the Greenstone interface to further annotate or edit the collection if desired. The musical data can also be published and maintained using Greenstone's many existing tools and features.

2. RELATED RESEARCH

There are a number of software packages for building digital libraries that can serve as alternatives to Greenstone, including both commercial and open source systems. Examples of the latter include DAITSS, DSpace, EPrints, Fedora and Keystone DLS. Marill and Lucza provide a discussion of their comparative merits [6]. Although many of these are excellent products, Greenstone has the particular advantage of a longstanding association with MIR research dating to the beginnings of the ISMIR conference.

There are also a number of audio feature extraction packages available that may be used as alternatives to jAudio, including Marsyas [9], MIRtoolbox [5] and Sonic Visualiser [4]. Although these are all excellent systems, jAudio has the special advantage of combining an easily extensible plug-in architecture for adding new features (as does Sonic Visualiser) with a cross-platform Java implementation.

There are also a few existing software platforms for mining a variety of Internet resources, such as Mozenda [15], and related research on integrating metadata is also being done in the semantic desktop community (e.g. NEPOMUK [17]). To the best of the authours' knowledge, however, jSongMiner is the only such software focusing specifically on music, and has the essential advantages of being both open source and specifically designed for integrating extracted data with digital repository software like Greenstone. The closest existing software is jMIR's jWebMiner [8], which focuses on extracting statistically-derived numerical features from the Internet, rather than the raw metadata mined by jSongMiner.

3. GREENSTONE

Greenstone [10] is an open-source and multilingual software suite for building and distributing digital library collections. A particular emphasis has been placed on promoting digital libraries in developing countries and in UNESCO's partner communities and institutions. Although Greenstone is intended for library collections that can consist of a wide and heterogeneous range of materials, not just music, it has certainly effectively been applied to musical collections in the past (e.g. in [2] and [3]).

A Greenstone library consists of one or more collections. These can each store many different types of documents, such as HTML files, PDFs, images, videos, audio files, MIDI files, etc. Each such document can be annotated with metadata tags, which can in turn be used to index, browse, search or otherwise organize or process a collection.

Given a set of documents, Greenstone can automatically build and link a collection, a process that can include the automated extraction of metadata as well as the creation of new documents. Greenstone comes packaged with a variety of such metadata extractors for different types of documents, and can be extended with *document plugins* for additional document types, as has been done here with jAudio and jSongMiner. For example, Greenstone can apply the CANTOR [1] optical music recognition tool to scans of scores as they are added to collections in order to automatically generate symbolic representations of the music.

Users can also use Greenstone to manually annotate resources with metadata using the *librarian's interface*. Greenstone collections can also be easily and automatically expanded by adding new documents to them.

Greenstone can publish digital libraries either to the Internet or to physical media such as CD-ROMs. The latter option is particularly important when working to make digital libraries accessible in locations where network access is limited or unavailable, such as in developing countries. The particular metadata fields that are published, as well as how they are formatted, are both highly configurable.

The Greenstone software and sample collections can be accessed at www.greenstone.org.

4. JMIR

jMIR [8] is a suite of software tools and other resources developed for use in automatic music classification research. jMIR includes the following components:

- **jAudio:** Extracts features from audio files.
- **jSymbolic:** Extracts features from symbolic music files.
- **jWebMiner 2.0:** Extracts statistical features from cultural and listener information available on the Internet.
- **jLyrics:** Extracts features from lyric transcriptions.
- **ACE 2.0:** A metalearning-based automatic classification engine.
- **jMusicMetaManager:** Software for managing and detecting errors in musical datasets and their metadata.
- **lyricFetcher:** Mines lyrics from the Internet.
- **jMIRUtilities:** Performs infrastructural tasks.
- **ACE XML:** Standardized MIR file formats.

- **Codaich, Bodhidharma MIDI and SAC/SLAC:** Musical research datasets.

All of the jMIR components emphasize extensibility, and they may be used both individually and as integrated groups. All jMIR components are open-source and are distributed free-of-charge at jmir.sourceforge.net.

5. EXTRACTING FEATURES FROM AUDIO DOCUMENTS IN GREENSTONE

As noted above, jAudio [7,8] is a jMIR component that extracts content-based features from audio files. A new jAudio Greenstone plugin has been implemented so that Greenstone can now automatically run jAudio to extract and store features from each audio file added to a Greenstone collection. jAudio itself has also been updated and expanded in order to make it easier to install and use, and to expand the range of codecs that it can use.

One way to take advantage of the features extracted by jAudio is to simply use them as descriptors, just like any other Greenstone metadata, something that can be particularly useful for higher-level features than have an explicit musical meaning. The extracted features may also be processed by classification software—such as jMIR ACE [8]—in order to arrive at still further metadata labels that can themselves be stored, such as content-derived predictions of labels like genre, mood, artist, etc.

jAudio can extract features from a variety of audio file formats, including MP3, FLAC, WAV, AIFF and AU. It is distributed with 28 base implemented features, including both low-level features (e.g. spectral flux and spectral centroid) and higher-level features (e.g. rhythmic features derived from beat histograms). This number of extracted features can be dramatically expanded at runtime, as jAudio includes *metafeatures* and *aggregators* [7,8] that can be used to automatically derive further features from base features, such as the standard deviation, rate of change or average of a given feature across a set of analysis windows.

In addition, one of the most important advantages of jAudio is that it is a relatively simple matter to add newly developed features using jAudio's plugin interface, without the need to recompile jAudio (or Greenstone). jAudio is also highly configurable, so users can decide which features to extract, whether or not to apply pre-processing like normalization or downsampling, etc.

Once features are extracted, they can simply be stored directly in the Greenstone collection metadata. They can also be exported as ACE XML [8] or Weka ARFF [11] files for external processing if desired.

6. USING JSONGMINER TO MINE METADATA

As noted above, jSongMiner is a novel software package that provides a framework for extracting metadata about musical entities from resources available on the Internet. Although it has been designed in the specific context of Greenstone, jSongMiner has been implemented such that it can also be used as a stand-alone application if desired, or used in conjunction with other jMIR components.

jSongMiner begins by identifying unknown audio files using audio fingerprinting (The Echonest's [14] fingerprinting services are used by default). jSongMiner can also identify recordings using metadata that is embedded in audio files or that is manually specified.

Once jSongMiner has identified a recording, it then extracts metadata about it from APIs offered by various online sources, or from metadata embedded in the audio file. jSongMiner keeps a record of resource identifiers in as many namespaces as possible while doing this, thus facilitating the integration of information from different sources.

In addition to collecting metadata about songs, jSongMiner can also automatically acquire metadata about artists and albums associated with songs. So, if given an unidentified song, jSongMiner will first identify it using audio fingerprinting, and then extract all available metadata on this song from all of the on-line resources that it has access to. If this metadata includes artist and/or album identifiers, then all available fields will also be extracted for this artist and/or album as well. In order to avoid redundant queries, jSongMiner can be set to only extract metadata on albums and artists for which it has not already extracted metadata.

jSongMiner thus allows users to treat songs, artists and albums as separate resource types, and allows information to be extracted and saved independently for each of them, whilst at the same time maintaining information outlining the connections between resources of the same and different types. Users also have the option of packaging artist and album metadata together with song metadata if they prefer.

Once metadata has been extracted relating to a song, artist and/or album, this metadata can be saved as an ACE XML [8] file or as a return-delimited text file. In the context of Greenstone, the jSongMiner Greenstone plugin allows all acquired data and metadata to be automatically incorporated into Greenstone's internal data structures. In any of these cases, jSongMiner allows the storage of metadata containing diverse character sets.

Each piece of metadata extracted by jSongMiner includes the field label, the metadata value and an identifier for the source from which the metadata was collected. The field labels are standardized, so that a given type of information will always be assigned the same field name by jSongMiner, regardless of where it is acquired from. For

example, jSongMiner will place the title of a song in the "Song Title" field, regardless of whether one data source might refer to it as "Song Name" and another as "Title".

The ability to identify the source of each piece of metadata is also important, as different sources might supply different results for a given field. For example, one source might identify the artist associated with a song as "Charles Mingus", and another might specify "Charlie Mingus". For this reason, jSongMiner allows multiple results for the same field to be extracted and stored in parallel. If the metadata is cleaned at some later point, the correction algorithm (or person) can be defined as a new source, and the original uncleaned metadata can be maintained or deleted, as desired. All of this means that jSongMiner organizes metadata from diverse sources in a structured and consistent way, whilst at the same time allowing any idiosyncrasies and subtleties implicit in the original data sources to be maintained and referenced if desired.

jSongMiner's ability to store multiple values for a given metadata field, from the same or different sources, also helps to make it possible to move beyond simple flat data structuring. This is enhanced by jSongMiner's (and ACE XML's) ability to link to external resources (including RDF ontologies) via metadata field entries, as well as by the way in which jSongMiner treats songs, artists and albums as distinct but linked entities.

Users can opt to have extracted metadata presented using unqualified or qualified Dublin Core [13] tags. In order to make this possible, jSongMiner includes original Dublin Core schemas. This use of Dublin Core can be particularly useful from a librarian's perspective.

The primary objective of jSongMiner is to provide a general framework that users can extend to incorporate whatever web services and data sources they wish. It was consciously decided not to design jSongMiner as a framework linked to any specific web services, as APIs change, web services go off-line and new ones appear. Furthermore, each on-line resource has its own terms of service potentially limiting which and how much data can be accessed and stored. A strong emphasis was therefore placed on designing jSongMiner in a modular way that allows it to be easily extended so that it can be used with arbitrary data sources, rather than biasing its architecture towards the APIs of any particular data sources.

So, one of the primary advantages of jSongMiner is the way in which it provides the basic extensible framework for incorporating functionality for accessing particular web services. Furthermore, it standardizes the ways that extracted metadata is labelled, structured and made accessible.

Having noted this, the decision was made to implement functionality for accessing data made available through the Echo Nest [14] and Last.FM [12] APIs, two of the richest sources of on-line metadata at the time of this writing. This was done primarily as a proof of concept and to make jSongMiner immediately useful out of the box.

Using the Echo Nest and Last.FM web services, jSongMiner can currently extract over one hundred song, artist and album metadata fields. In addition, many of these fields can have multiple values. For example, there will usually be multiple artists listed in the "Similar Artist" field.

The jSongMiner fields range from standard musical fields (e.g. "Song Title" or "Genre") to primary keys (e.g. "Echo Nest Song ID" or "Music Brainz Artist ID") to content-based information (e.g. "Duration (seconds)" or "Tempo (BPM)") to consumption-based data (e.g. "Last.FM Track Play Count" or "Echo Nest Artist Hotness (0 to 1)") to links to external textual data (e.g. "Artist-Related Blog" or "Last.FM Album Wiki Text") to links to multimedia (e.g. "Artist-Related Image" or "Artist-Related Video").

In order to make jSongMiner as flexible as possible, the software is highly customizable in terms of what kinds of information are extracted, where it is extracted from and how the data is structured. Such options can be set through jSongMiner's configuration files and its command line.

Every effort has been made to make jSongMiner as easy to use as possible, with ample documentation in the manual, so even users with only moderate computer backgrounds should still have relatively little difficulty using the software. In addition to including a command line and configuration file-based interface that makes the jSongMiner easy to run from other software, jSongMiner also has a well-documented API in order to facilitate the use of jSongMiner as a library incorporated into other software.

If the jSongMiner Greenstone plugin is being used, then the user never needs to interact with jSongMiner directly while using Greenstone. The plugin simply has jSongMiner perform tasks in the background, and data it extracts is automatically structured and linked within the collection produced by Greenstone. jSongMiner configuration settings can also be specified within the Greenstone interface.

Like all jMIR components, jSongMiner is cross-platform, open-source and available for free at jmir.sourceforge.net.

7. USE CASES AND CASE STUDIES

Greenstone is designed to be used for a variety of different musical purposes by a variety of user types. This section briefly describes a few of the many possible use cases.

MIR researchers, especially those specializing in music classification, are the first user group that will be considered. Such researchers often have a need for datasets that can be used to evaluate and compare algorithms. These datasets should ideally also be well-annotated with metadata

that can, among other things, serve as class labels. Greenstone could be used by those building MIR research datasets not only to harvest rich metadata about their music files, but also to export and publish information about the dataset to the web as linked HTML that other researchers could search and browse when choosing a dataset to use in their own research. It should be emphasized that Greenstone's ability to extract content-based features is especially useful in this context, as this facilitates the publication and distribution of a dataset's extracted features even when the music itself cannot be distributed due to legal limitations.

To serve as an example, a Greenstone collection was generated from the audio files of SAC/SLAC, a research dataset that has been used in a number of previous studies (e.g. [8]). Greenstone automatically extracted content-based features and mined metadata from the web, as described above. The result is an automatically annotated Greenstone collection, whose metadata can be browsed, searched, edited and published. Figure 1 shows a screen shot of one sample entry. The full published Greenstone collection is posted at www.nzdl.org/greenstone2-jmir.

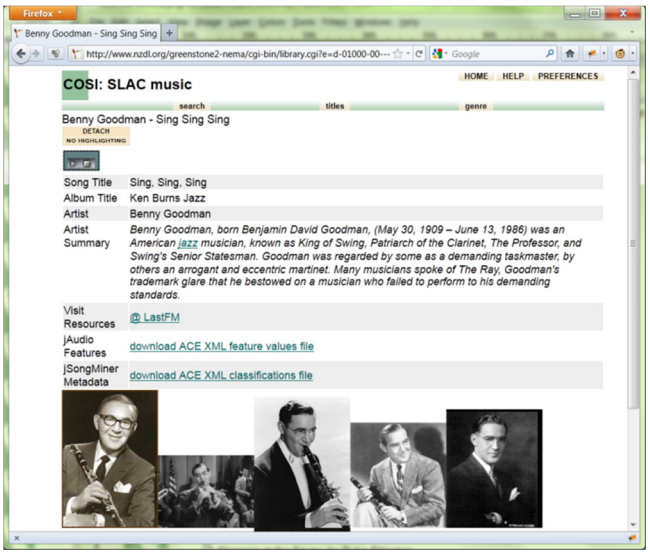

Figure 1: A sample entry on *Sing, Sing, Sing,* by Benny Goodman, from the COSI-SLAC Greenstone research collection. Under the particular display configuration settings that were chosen for this collection, the main entry displays only basic summary information, and the audio features and full detailed metadata are left to be downloaded as ACE XML files for machine processing or viewing. Mined images are also displayed, and the audio itself is streamed.

Considered from a somewhat different perspective, some of the metadata mined by jSongMiner can also be used directly as features, even though this is not its primary purpose (e.g. "Tempo BPM", "Key", "Time Signature", etc.). This type of usage is facilitated by jSongMiner's (and Greenstone's) ability to save metadata in ACE XML [8], a machine learning-oriented format.

The SLAC collection was also used to investigate this application experimentally. jSongMiner identified each of the audio recordings in SLAC using fingerprinting, and was then used in combination with jMIR's jWebMiner [8] in order to mine a variety of features using the APIs of Last.FM and Yahoo. These features were then used to perform a 10-class 10-fold genre classification experiment, where jMIR's ACE [8] provided the machine learning functionality. This resulted in an average 83% classification success rate, compared to 68% when only audio content-based features extracted by jAudio were used. For the sake of comparison, 86% was achieved when the same web-derived features were used, but model curated identifiers were used to extract them rather than identifiers derived from jSongMiner's fingerprinting results.

Music librarians are another important potential user of the updated Greenstone software. Even those libraries with extensive digital collections tend to have relatively limited metadata available for the bulk of their collections. The cost of manually annotating music is a major stumbling block, and Greenstone now allows the process to be cheaply and easily automated. Librarians simply need to provide music to Greenstone, which will then automatically annotate it with metadata. Librarians can then validate the extracted metadata if they wish, a process much cheaper than actually entering it. The metadata can then be published to the web or CD using Greenstone to provide increased access to library patrons, and the Dublin Core tags generated by Greenstone can be used for internal reference purposes.

There are also many other potential user types. Private music collectors might wish to use Greenstone to annotate their collections, for example, or to detect wrongly labelled recordings using fingerprinting. To give another example, those in the music industry might use it to enrich their own catalogue or marketing data in a variety of ways. It is especially important to emphasize that jSongMiner is designed to be easily extended to mine data using arbitrary APIs, so there may be many types of data which could potentially be accessed in the future which have not been envisioned yet.

8. CONCLUSIONS AND FUTURE RESEARCH

The incorporation of the updated jAudio and the new jSongMiner software into Greenstone significantly expands Greenstone's value to those wishing to automatically construct, annotate, organize and make accessible large music collections. Greenstone can now identify unknown audio recordings, extract content-based information from

audio files and mine Internet resources in order to automatically build a rich set of metadata about musical entities. Such Greenstone collections can consist of many different types of documents associated with each musical piece, artist or album, such as audio files, scores, videos, images and PDFs. Users also have the ability to use jSongMiner or jAudio outside of the Greenstone framework if they wish.

One of the main priorities of future research is to more fully incorporate Greenstone and jMIR into the Networked Environment for Music Analysis [16] project. This will allow Greenstone collections to be built and accessed in a distributed framework that will further increase its usefulness to MIR researchers.

Another priority is the design of Greenstone plugins for other jMIR components, so that, for example, features may also be automatically extracted from MIDI files added to a Greenstone collection using jSymbolic, or from lyrical transcriptions using jLyrics.

An additional priority is the direct incorporation of further web services into jSongMiner. Although the main value of jSongMiner is as a framework that facilitates the incorporation of arbitrary web services as they become available, it would still be advantageous for some potential users to build in immediate support for further currently existing on-line resources, such as MusicBrainz, Yahoo! and Amazon, to name just a few.

The fourth priority is the integration of functionality for automatically detecting errors in collected metadata. The already existing functionality in jMIR's jMusicMetaManager [8] will be a good starting point. This functionality will also ideally be expanded to perform auto-correction that can automatically update the data sources from which erroneous metadata was mined, if permitted.

9. ACKNOWLEDGEMENTS

The authors would like to thank the Centre for Open Software Innovation (COSI) and the Andrew W. Mellon Foundation for their generous financial support. Thanks also to Daniel McEnnis for his work on jAudio, and to the many others who have contributed to jMIR or Greenstone in the past, especially Prof. Ichiro Fujinaga.

10. REFERENCES

[1] Bainbridge, D., and T. Bell. 2003. A music notation construction engine for optical music recognition. *Software Practice and Experience* 33 (2): 173–200.

[2] Bainbridge, D., S. J. Cunningham, and J. S. Downie. 2004. GREENSTONE as a music digital library toolkit. *Proceedings of the International Conference on Music Information Retrieval*. 42–7.

[3] Bainbridge, D., S. J. Cunningham, and J. S. Downie. 2004. Visual collaging of music in a digital library. *Proceedings of the International Conference on Music Information Retrieval*. 397–402.

[4] Cannam, C., C. Landone, M. Sandler, and J. P. Bello. 2006. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. *Proceedings of the International Conference on Music Information Retrieval*. 324–7.

[5] Lartillot, O., and P. Toiviainen. 2007. MIR in Matlab (II): A toolbox for musical feature extraction from audio. *Proceedings of the International Conference on Music Information Retrieval*. 127–30.

[6] Marill, J. L., and E. C. Lucza. 2009. Evaluation of digital repository software at the National Library of Medicine. *D-Lib Magazine* 15 (5/6).

[7] McEnnis, D., C. McKay, and I. Fujinaga. 2006. jAudio: Additions and improvements. *Proceedings of the International Conference on Music Information Retrieval*. 385–6.

[8] McKay, C. 2010. Automatic music classification with jMIR. *Ph.D. Dissertation.* McGill University, Canada.

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

[10] Witten, I. H., D. Bainbridge, and D. M. Nichols. 2010. *How to build a digital library.* San Francisco, CA: Morgan Kaufmann.

[11] Witten, I. H., and E. Frank. 2005. *Data mining: Practical machine learning tools and techniques*. New York: Morgan Kaufman.

[12] API – Last.fm. Retrieved 15 August 2011, from http://www.last.fm/api.

[13] Dublin Core Metadata Initiative. Retrieved 22 August 2011, from http://dublincore.org.

[14] Echo Nest API Overview. Retrieved 15 August 2011, from http://developer.echonest.com/docs/v4/.

[15] Mozenda. Retrieved 15 August 2011, from http://www.mozenda.com.

[16] Networked Environment for Music Analysis (NEMA). Retrieved 15 August 2011, from http://www.music-ir.org/?q=nema/overview.

[17] Semantic Desktop with KDE. Retrieved 15 August 2011, from http://nepomuk.kde.org.