DISCOVERING METADATA INCONSISTENCIES

Bruno Angeles
CIRMMT
Schulich School of Music
McGill University
bruno.angeles@mail.mcgill.ca

Cory McKay
CIRMMT
Schulich School of Music
McGill University
cory.mckay@mail.mcgill.ca

Ichiro Fujinaga
CIRMMT
Schulich School of Music
McGill University
ich@music.mcgill.ca

ABSTRACT

This paper describes the use of fingerprinting-based querying in identifying metadata inconsistencies in music libraries, as well as the updates to the jMusicMetaManager software in order to perform the analysis. Test results are presented for both the Codaich database and a generic library of unprocessed metadata. Statistics were computed in order to evaluate the differences between a manually-maintained library and an unprocessed collection when comparing metadata with values on a MusicBrainz server queried by fingerprinting.

1. INTRODUCTION

1.1 Purpose

Metadata is useful in organizing information, but in large collections of data it can be tedious to keep that information consistent. Whereas decision making in previous environments such as traditional libraries was limited to a small number of highly trained and meticulous people, the democratization of music brought about by the digital age poses new challenges in terms of metadata maintenance, as music can now be obtained from diverse and potentially noisy sources.

The contributors to many popular metadata repositories tend to be much less meticulous, and may have limited expertise. The research presented here proposes a combination of metadata management software, acoustic fingerprinting, and the querying of a metadata database in order to discover possible errors and inconsistencies in a local music library.

Metadata entries are compared between a library of manually-maintained files and a metadata repository, as well as between a collection of unprocessed metadata and the same repository, in order to highlight the possible differences between the two.

1.2 Metadata and its Value

Metadata is information about data. In music, it is information related to recordings and their electronic version (such as the performer, recording studio, or lyrics), although it can also be event information about artists or other attributes not immediately linked to a recorded piece of music. Corthaut et al. present 21 semantically related clusters of metadata [1], covering a wide range of information that illustrates the variety of metadata that can be found in music. Lai and Fujinaga [6] suggest more than 170 metadata elements organized into five types, in research pertaining to metadata for phonograph recordings. Casey et al. [2] distinguish factual from cultural metadata. The most widely used implementation of musical metadata is the ID3 format associated with MP3 files [5].

The main problem with metadata is its inconsistency. The fact that it is stored in databases containing thousands, if not millions, of entries often means that the data is supplied by several people who may have different approaches. Spelling mistakes may go unnoticed for a long time, and information such as an artist's name might be spelled in different but equally valid ways. Additionally, several metadata labels—most notably *genre*—are highly subjective.

When maintaining large databases of music, valid and consistent metadata facilitates the retrieval and classification of recordings, be it for Music Information Retrieval (MIR) purposes or simply for playback.

1.3 Metadata Repositories and Maintenance Software

Metadata repositories are databases that include information about recordings. When they are supported by an Application Programming Interface (API), they provide users with a convenient way of accessing the stored metadata. Existing music metadata repositories include MusicBrainz, Discogs, Last.fm, and Allmusic, to name just a few [3].

Several software solutions exist that provide ways to access, use, and adjust musical metadata. These include MusicBrainz Picard, MediaMonkey, jMusicMetaManager, and Mp3tag. The first three applications support fingerprinting in some form, but Mp3tag does not. GNAT [10] allows querying by metadata or fingerprinting to explore aggregated semantic Web data. jMusicMetaManager is the only application that performs extensive automated internal textual metadata error detection, and it also produces numerous useful summary statistics not made available by the alternatives.

1.4 Acoustic Fingerprinting

Acoustic fingerprinting is a procedure in which audio recordings are automatically analyzed and deterministically associated with a key that consumes considerably less space than the original recording. The purpose of using the key in our context is to retrieve metadata for a given recording using only the audio information, which is more reliable than using the often highly noisy metadata packaged with recordings.

Among other attributes, fingerprinting algorithms are distinguished by their execution speed; their robustness to noise and to various types of filtering; and their transparency to encoding formats and associated compression schemes [1].

The fingerprinting service used in this paper is that of MusicIP (now known as AmpliFIND). It is based on Portable Unique Identifier (PUID) codes [9]. These are computed using the GenPUID piece of software. The PUID format was chosen for its association with the MusicBrainz API.

1.5 Method

This research uses jMusicMetaManager [7] (a Java application for maintaining metadata), Codaich [7] (a database of music with manually-maintained metadata), a reference library of music labeled with unprocessed metadata, and a local MusicBrainz server at McGill University's Music Technology Area. Reports were generated in jMusicMetaManager, an application for music metadata maintenance which was improved as part of this project by the addition of fingerprinting-based querying. This was done in order to find the percentage of metadata that was identical between the manually-maintained metadata and that found on the MusicBrainz server of metadata. In addition to comparing the *artist*, *album*, and *title* fields, a statistic was computed indicating how often all three of these specific fields matched between the local library and the metadata server, a statistic that we refer to as "identical metadata." Raimond et al. [10] present a similar method, with the ultimate objective of accessing information on the Semantic Web.

An unprocessed test collection, consisting of music files obtained from file sharing services, was used in order to provide a comparison between unmaintained and manually-maintained metadata. This unprocessed library is referred to as the "reference library" in this paper.

2. JMUSICMETAMANAGER

jMusicMetaManager [7] is a piece of software designed to automatically detect metadata inconsistencies and errors in musical collections, as well as generate descriptive profiling statistics about such collections. The software is part of the jMIR [8] music information retrieval software suite, which also includes audio, MIDI, and cultural feature extractors; metalearning machine learning software; and research datasets. jMusicMetaManager is, like all of the jMIR software, free, open-source, and designed to be easy to use.

One of the important problems that jMusicMetaManager deals with is the inconsistencies and redundancies caused by multiple spellings that are often found for entries that should be identical. For example, uncorrected occurrences of both "Lynyrd Skynyrd" and "Leonard Skinard" or of the multiple valid spellings of composers such as "Stravinsky" would be problematic for an artist identification system that would incorrectly perceive them as different artists.

At its simplest level, jMusicMetaManager calculates the Levenshtein (edit) distance between each pair of entries for a given field. A threshold is then used to determine whether two entries are likely to, in fact, correspond to the same true value. This threshold is dynamically weighted by the length of the strings. This is done separately once each for the *artist, composer, title, album,* and *genre* fields. In the case of titles, recording length is also considered, as two recordings might correctly have the same title but be performed entirely differently

This approach, while helpful, is too simplistic to detect the full range of problems that one finds in practice. Additional pre-processing was therefore implemented and additional post-modification distances were calculated. This was done in order to reduce the edit distance of strings that probably refer to the same thing, thus making it easier to detect the corresponding inconsistency. For example:

- Occurences of "The " were removed (e.g., "The Police" should match "Police").
- Occurrences of " and " were replaced with " & "
- Personal titles were converted to abbreviations (e.g., "Doctor John" to "Dr. John").
- Instances of "in'" were replaced with "ing" (e.g., "Breakin' Down" to "Breaking Down").
- Punctuation and brackets were removed (e.g., "R.E.M." to "REM").
- Track numbers from the beginnings of titles and extra spaces were removed.

In all, jMusicMetaManager can perform 23 pre-processing operations. Furthermore, an additional type of processing can be performed where word orders are rearranged (e.g., "Ella Fitzgerald" should match "Fitzgerald, Ella," and "Django Reinhardt & Stéphane Grappelli" should match "Stéphane Grappelli & Django Reinhardt"). Word subsets can also be considered (e.g., "Duke Ellington" might match "Duke Ellington & His Orchestra").

jMusicMetaManager also automatically generates a variety of HTML-formatted statistical reports about music collections, including multiple data summary views and analyses of co-occurrences between artists, composers, albums, and genres. This allows one to easily acquire and publish HTML collection profiles. A total of 39 different HTML reports can be automatically generated to help profile and publish musical datasets.

Users often need a graphical interface for viewing and editing a database's metadata. It was therefore decided to link jMusicMetaManager to the Apple iTunes software, which is not only free, well-designed, and commonly used, but also includes an easily parsed XML-based file format. iTunes, in addition, has the important advantage that it saves metadata modifications directly to the ID3 tags of MP3s as well as to its own files, which means that the recordings can easily be disassociated from iTunes if needed. iTunes can also access Gracenote's metadata automatically, which can then be cleaned with jMusicMetaManager.

jMusicMetaManager can extract metadata from iTunes XML files as well as directly from MP3 ID3 tags. Since Music Information Retrieval systems do not typically read these formats, jMusicMetaManager can also be used to generate ground-truth data formatted in ACE XML or Weka ARFF formats.

3. CODAICH

Codaich is a curated audio research dataset that is also part of jMIR [8]. It is constantly growing, and is now significantly larger than its original size of 20,849 recordings. The version used for the experiments described in this paper contains 32,328 recordings.

There is music by nearly 3,000 artists in Codaich, with 57 different musical genres represented. The dataset can be divided into four broad genres of Jazz, Popular, Classical, and (the somewhat problematic) World, henceforth referred to as "genre groups." These recordings are labeled with 19 metadata fields.

The metadata of the original version of Codaich was meticulously cleaned, both by hand and with jMusic-MetaManager. Care has been taken to maintain this very high level of metadata quality as the dataset has grown. The metadata for the original version of Codaich is available at the jMIR web site (http://jmir.sourceforge.net), and the metadata of the most recent version can be obtained in iTunes XML form by contacting the authors.

4. THE REFERENCE LIBRARY

In order to provide context, it was decided that a benchmark was needed against which the metadata consistency between Codaich and MusicBrainz could be compared. This was the motivation behind assembling the reference library, a combination of files downloaded from torrent-based networks and files that were obtained before the emergence of such systems. In the former case, files were downloaded as entire albums, while the rest of the reference library consists of recordings that were downloaded individually. The reference library consists of 1363 recordings by 446 artists, with 70 musical genres represented.

Since the reference library contained many music files with no ID3 metadata, but did hold some information in the files' names, metadata was created in such cases based on file names.

5. METHOD

5.1 Overview of the Experiments

Experiments were conducted to determine whether or not manually-maintained Codaich musical metadata showed a different level of consistency with MusicBrainz's information than the unprocessed reference library, for a fixed number of metadata fields.

The first step of the experiments consisted of obtaining PUID codes for each recording in each of the two libraries. The PUID information was stored in an XML file for later parsing by jMusicMetaManager.

In jMusicMetaManager, all the recordings in Codaich and the reference library were matched with entries in the XML file of PUID codes. PUID-based querying was performed on the MusicBrainz server, and a report of matching fields was generated for the chosen metadata fields.

Similar research done by Raimond et al. [10] presents GNAT, a Python application that supports PUID-based fingerprinting for track identification on a personal music library. The authors suggest accessing information pertinent to the user through the Semantic Web by querying, while we analyze the rate of consistency between the two datasets.

5.2 Changes to jMusicMetaManager

Running jMusicMetaManager on a large library of music files revealed that the application was not able to read the ID3 tags of files using releases 1, 2, 3, and 4 of the ID3v2 protocol. This was due to the choice of metadata API, *java_mp3*, used in jMusicMetamanager. Replacing *java_mp3* with the *jaudiotagger* API allowed us to read those formats.

Fingerprinting-based querying was added to jMusicMetaManager to enhance its capabilities. MusicBrainz's official (although no longer in active development) Java API was used (it is known as *libmusicbrainz-java*), since it allows querying by PUID, and a new corresponding report was added to jMusicMetaManager.

To expedite the querying process, threaded querying was implemented. This was applied to a copy of the MusicBrainz database hosted on a server at McGill University, something that was important in overcoming the one-query-per-second limitation of the public MusicBrainz server.

Genre group	Number of identified recordings	Identical artist	Identical album	Identical title	Identical artist, album, and title
Classical	1,476	3%	2%	6%	0%
Jazz	3,179	70%	25%	**64%**	12%
Popular	16,206	**84%**	**52%**	61%	**32%**
World	1,640	58%	29%	46%	11%

Table 1. Querying results for Codaich. Percentages represent the number of entries that were identical to those in MusicBrainz. The top results per statistic are identified in bold.

Genre group	Number of identified recordings	Identical artist	Identical album	Identical title	Identical artist, album, and title
Classical	285	17%	0%	5%	0%
Jazz	181	43%	14%	39%	4%
Popular	481	**79%**	**19%**	**51%**	**10%**
World	115	57%	12%	41%	3%

Table 2. Querying results for the reference library. Percentages represent the number of entries that were identical to those in MusicBrainz. The top results per statistic are identified in bold.

Genre group	Identical artist	Identical album	Identical title	Identical artist, album, and title
Classical	-14%	2%	1%	0%
Jazz	**27%**	11%	**25%**	8%
Popular	5%	**33%**	11%	**22%**
World	2%	17%	6%	9%

5.3 Reporting and Statistics

Not all files listed in the XML file of PUID codes were successfully identified by the MusicBrainz server (and, of course, MusicBrainz identification does not guarantee correctness). Several files list *unanalyzable* or *pending* as their status, while other extracted PUID codes did n return any result at all. Only identified recordings are used in this paper's statistics. The ratios of files that were not processed in each collection are specified in the following sections.

A case-insensitive string comparison was used in order to determine whether or not the *artist*, *album,* and *title* fields were identical on the MusicBrainz server and in the files' metadata.

6. RESULTS

Reports were generated for both Codaich and the reference library. The former database is maintained manually and is assumed to contain very few metadata errors and inconsistencies, while the latter contains many metadata problems due to the wide range of contributors and their varied interest and methods in maintaining metadata.

6.1 Codaich Results

Of the 32,328 songs in Codaich, 22,501 (70%) were identified on the MusicBrainz server using PUID values, 44 files were assigned a status of *unanalyzable* by GenPUID, and 84 were assigned the label *pending*. Of the remaining files, 9,645 (30%) had a PUID value but resulted in no hit on the MusicBrainz server.

Table 1 shows the metadata consistency between Codaich and MusicBrainz.

6.2 Reference Library Results

Of the 1,363 songs in the reference library, 1,062 (78%) were identified on the MusicBrainz server using PUID values. 5 files were assigned a status of *unanalyzable* by GenPUID, and 18 were assigned the label *pending*. Of the remaining files, 274 (20%) had a PUID value but resulted in no hit on the MusicBrainz server.

Table 2 shows the metadata consistency between the reference library and MusicBrainz.

6.3 Comparison of Codaich and the Reference Library

Table 3 illustrates the difference between the entries of Table 2 and Table 1. Positive values indicate a higher rate of matching metadata between MusicBrainz and Codaich than between MusicBrainz and the reference library, while negative values mean the opposite. Although the first two tables are based on different libraries, the values of their difference provide us with a rough estimate of the quality difference between metadata in unprocessed music files collected from the internet and a curated library.

7. DISCUSSION

7.1 Global Observations

A comparison of Table 1 and Table 2 shows that the strongest agreement with MusicBrainz for the *artist* and *album* fields, as well as for the "identical metadata" statistic, is obtained for Popular music. This supports the argument that the main drivers of community-based metadata services are musical genres with which the most people are familiar, particularly among the technologically-savvy younger generations who may be more likely to contribute to metadata libraries.

With respect to titles, however, there is a greater level of MusicBrainz consistency in the manually-maintained library for Jazz recordings than for Popular recordings (albeit only 3% more, and the MusicBrainz title consistency is greatest for Popular music in the reference library). This may be due to better knowledge of Jazz on the part of Codaich's curator relative to the general public.

A potential cause of the relatively low percentages of Table 2 is the fact that part of the reference library consists of files that used ID3v1 tags instead of ID3v2 ones. ID3v1 tags limit the size of the *title*, *artist*, and *album* fields to 30 characters [5], which could cause mismatches in the case of longer entries of the MusicBrainz server compared to the limited ones of the ID3v1 tags.

7.2 Differences Between a Curated Database and an Unprocessed Collection

The largest changes between the two collections, as can be seen from Table 3, were obtained (in decreasing order) for the *album* field of Popular music (33%), the *artist* field of Jazz (27%), the *title* field of Jazz (25%), and all three chosen fields for Popular music (22%). In 14 out of 16 statistics, the MusicBrainz metadata matches the curated database's information more often than it does the unprocessed collection. We now focus on the two fields that were more MusicBrainz-consistent for the reference library than for Codaich. For Classical music, the consistency of the *artist* field for the reference library (17%) is much higher than that for Codaich (3%).

We were surprised to notice that the *artist* field results appeared to be "worsened" by the manual maintenance of metadata in this way. This can perhaps be explained by noting that Codaich fills the *artist* field with the name of the performer, while reserving composer names for the *composer* field. Most metadata on the MusicBrainz server and in the reference library, in contrast, lists the composer's name in the *artist* field, ignoring the *composer* field, possibly due to inherent limitations of the choice of underlying database schema.

The second statistic that is not higher in Table 1 than in Table 2 is "identical metadata" (artist, album, and title) for Classical music. In both cases, this statistic has a value of 0%, meaning that none of the Classical files considered had values matching the MusicBrainz entries for all three of these fields.

Indeed, the MusicBrainz consistency for Classical Music was very low, even for individual fields. Associating metadata with Classical music is challenging, as one must make decisions such as choosing how to convert names from the Cyrillic alphabet to Latin characters[1], choosing who the artist is, choosing whether to include long subtitles in album names, choosing whether to include the key and opus number in the title, etc. It is important to note that different ways of writing metadata may be perfectly valid in these situations, but multiple valid values can cause retrieval problems.

7.3 Classical/World Music vs. Jazz/Popular Music

Tables 1 and 2 allow us to distinguish two sets of genre groups, based on the frequency of matching metadata between the local files and the MusicBrainz server: Jazz and Popular music in one group, World and Classical music in the other.

Indeed, the highest matches are most often obtained for Jazz and Popular music, while the lowest results are in general obtained for Classical and World music. Popular and Classical music have different characteristics and use different fields [4], which leads to complications when applying a uniform metadata structure to different genres.

Classical music has by far the lowest MusicBrainz agreement in both music collections. World music has results between those of Classical music and Popular/Jazz music. The fact that Popular (and to a certain extent Jazz) music uses clearly-defined *artist*, *album,* and *title* tags facilitates the matching of such information on a web server.

7.4 Matching Results and Correct Results

Human maintenance of metadata has the expected effect of making metadata consistent across a library. Let us consider the case of Classical music. The low rate of consistency of matching artist names between Codaich and MusicBrainz might lead us to think that manual maintenance had a negative effect on the metadata, but in reality the Codaich metadata is arguably better than the

[1] Although this happens in other genres, it could be argued that conversion between languages is statistically more likely in Classical and World music.

MusicBrainz metadata. The ID3v2 protocols support a *composer* field, and should be used for that purpose, as done in Codaich.

It was interesting to observe in Codaich's file-by-file report of metadata comparisons that certain files that were assumed to belong to the same album were listed as belonging to different ones in the web service's fetched metadata. Consider the case of recordings that appear in different contexts such as movie soundtracks, compilations, and regular studio albums. Certain users may want to keep these recordings identified as being associated with the original studio release, while others will prefer to associate them with the other releases on which they appear, both points of view being valid. Such an issue would be avoided by allowing multi-value metadata fields.

8. CONCLUSIONS

Our results indicate that manually-maintained music files tend to have a greater level of metadata consistency with MusicBrainz than do unprocessed files. This does not, however, mean that the web service's metadata contains the correct values. We also noted that the matching rates of metadata vary across the analyzed genre groups. Differences in metadata between a manually-maintained database and a metadata database such as MusicBrainz may be due to a variety of reasons.

Combining jMusicMetaManager with fingerprinting querying can facilitate the cleanup of local collections of music. The matching of metadata between local files and a metadata server is particularly useful in the case of Popular music and Jazz, recent genres for which metadata fields are more easily attributed than for Classical and World music. With this new querying feature, jMusicMetaManager is useful in more situations and unique in its reporting capability. We have also seen that, although at first sight the manual maintenance of metadata revealed some lowering of matches with the MusicBrainz server, it was justified by the proper use of attributes by the human curator. This can be seen as an illustration of the unreliability of collaboratively-maintained databases such as MusicBrainz for musical genres that do not benefit from the same public exposure as Popular music and Jazz.

In light of this potential unreliability, the use of metadata management software such as jMusicMetaManager is recommended in order to detect potential errors and inconsistencies in local libraries of music, as it can detect problems without reference to external sources of potentially unreliable information.

Having discussed the possibility that multiple values of a metadata field may all be valid in certain cases, we stress the need for multidimensional musical metadata.

Through our analysis of statistical results, we confirm the pertinence of the manual maintenance of metadata, and explain the reasons behind minor unexpected results.

jMusicMetaManager already computes metrics that can be used to detect metadata inconsistencies. As future work, integrating such features in the comparison of local and remote metadata would be helpful in that a threshold of comparison could allow the user to identify metadata that is similar enough. We expect an improvement in Classical music retrieval with such a change.

Finally, a similar experiment could be performed by manually correcting a given library of music while keeping the unprocessed version as reference.

9. REFERENCES

[1] Cano, P., E. Battle, T. Kalker, and J. Haitsma. 2002. A Review of Algorithms for Audio Fingerprinting. *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 169–73.

[2] Casey, M., R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. 2008. Content-Based Music Information Retrieval. *Proceedings of the IEEE*. 668–96, 2008.

[3] Corthaut, N., S. Govaerts, K. Verbert, and E. Duval. 2008. Connecting the dots: Music metadata generation, schemas and applications," *Proceedings of the International Conference on Music Information Retrieval*. 249–54.

[4] Datta, D. 2002. Managing metadata, *Proceedings of the International Conference on Music Information Retrieval*. 249–51.

[5] ID3.org. 2003. Home: ID3.org. http://www.id3.org. Accessed 21 March 2010.

[6] Lai, C., and I. Fujinaga. 2006. Data dictionary: Metadata for phonograph records. *Proceedings of the International Conference on Music Information Retrieval.* 1–6.

[7] McKay, C., D. McEnnis, and I. Fujinaga. 2006. A large publicly accessible prototype audio database for music research," *Proceedings of the International Conference on Music Information Retrieval*. 160–3.

[8] McKay, C., and I. Fujinaga. 2009. jMIR: Tools for automatic music classification," *Proceedings of the International Computer Music Conference*. 65–8.

[9] MusicBrainz. 2009. How PUIDs work: MusicBrainz Wiki. http://wiki.musicbrainz.org/ HowPUIDsWork. Accessed 21 March 2010.

[10] Raimond, Y., C. Sutton, and M. Sandler. 2008. Automatic interlinking of music datasets on the semantic web. *Linked Data on the Web Workshop (LDOW2008).*