GRAIL: A MUSIC METADATA IDENTITY API

Michael D. Barone[1] **Kurt Dacosta**[1] **Gabriel Vigliensoni**[2]
Matthew H. Woolhouse[1]
[1] McMaster University, Canada
[2] McGill University, Canada
baronem@mcmaster.ca, dacostk@mcmaster.ca, gabriel@music.mcgill.ca, woolhouse@mcmaster.ca

ABSTRACT

Music metadata curation can be a daunting task due to the large amount of existing information and varying degrees of organizational quality. Linking multiple music identities (IDs) can be beneficial for multi-disciplinary research—a combination of different resources can lead to enriched knowledge. Although ID linkage appears to have been solved to a degree in commercial contexts, there are still no open-access tools available for academic music research at the three core entities of music items: artists, releases, and tracks. The General Recorded Audio Identity Linker (GRAIL) is a music metadata ID linking API that (i) accepts many types of seed data to begin its linkage process, (ii) verifies and links music metadata IDs by means of a set of strict criteria, (iii) confirms linkage consistency using continuous crawling of music-service APIs, and (iv) provides ID linkages as a free, publicly available resource. To date, more than 30 million tracks, 2.7 million releases (albums), and 28 million artists IDs have been ingested into GRAIL, making it possibly one of the largest open-access music metadata resources available to date. A demonstration of the API will be available during the late-breaking session.

1. MOTIVATION

Music metadata resources, such as The Echo Nest, have been useful for current research within MIR [2, 7]. However, utilizing these APIs for research and development can be difficult because services often deprecate shortly after launch, or become restricted due to private acquisition. The acquisition of the Compact Disc Database [1], and the recent downgrading of the Echo Nest API [2] exemplify some of the challenges of using this information. Due to these challenges, research integrating multiple resources is relatively scarce. Where it does exist, convoluted string matching procedures can be required to improve linkage accuracy [8]. Furthermore, the quality of the metadata can lack integrity and reliability [4]. Despite these challenges, research that leverages data available from multiple resources can be powerful; the most accurate automated genre classification utilizes a combination of lyrical, cultural, and acoustic metadata [5].

Other research has examined the current issue, and best practices for music metadata organization [6]. For example, The Music Information Retrieval Evaluation Exchange (MIREX) [3] share large sums of digitally archived audio for MIR specialists interested in signal processing. However, to our knowledge, there are no resources that provide simple mapping of ID spaces to a large set of music metadata resources. The motivation of GRAIL, therefore, is to provide as an open-access music data-linkage service in in an attempt to ameliorate some of the inconsistencies as outlined in [1]. GRAIL:

- Automatically validates linkage strength
- Provides detailed documentation validation process
- Can assign multiple IDs to a single track
- Prioritizes correct, complete, and consistent linkages
- Does not depend upon the ontology of existing resources
- Is accessible as an open-access API

2. IMPLEMENTATION

GRAIL is an open framework, which can accept any type of seed data (e.g. track title, universal identifier), and return a set of linked music metadata IDs for that seed. Based on research needs, users can select linked IDs with more, or less secure linkages to fulfill the task. Linkages include confidence values which are generated via a set validation criteria. Validation criteria are not universal for every resource, and are based on the most consistent and reliable information available from the seed data, We describe the APIs architecture using our International Standard Recording Code (ISRC)-Spotify-MusicBrainz linkage process (Figure 1). To date, GRAIL contains linked metadata for 17 million songs from 14 active/inactive metadata providers (see Table 1).

Step 1 - Seed Data 27 million ISRCs linked to track-level catalog information were made available to our team as part of a data-sharing agreement with Nokia Music. The

[1] http://web.archive.org/web/20081227004501/http://www.wired.com/entertainment/music/commentary/listeningpost/2006/11/72105?currentPage=all
[2] http://developer.echonest.com/

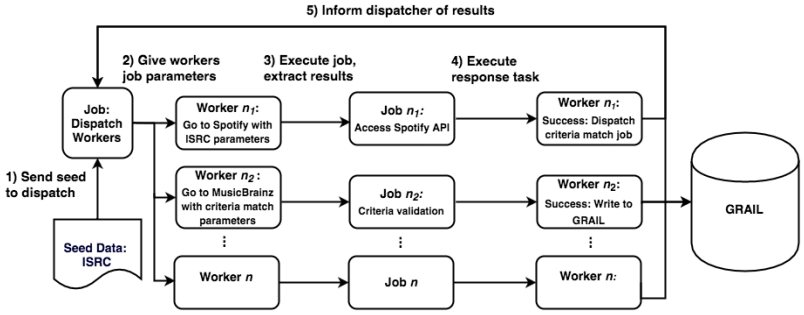

Figure 1: Spotify-to-MusicBrainz linkage workflow

unique mapping of ISRCs to Nokia Music is used as a starting point for GRAIL [3].

Step 2 - Dispatcher The dispatcher contains jobs which a worker can perform. Jobs include database writing, API querying, and linkage validation procedures. ISRCs are sent to the dispatcher as seed data. The job dispatch spawns n workers. Each worker is given specific a job, including all necessary parameters. For example, a worker receives an ISRC and a Spotify API endpoint [4]. Worker n_1 requests track data from Spotify using an ISRC. Worker n_2 receives a MusicBrainz end point with data parsed from a successful Spotify job performed by Worker n_1 in a successive iteration.

Step 3 - Execute job The worker evaluates the response from their executed job and returns a response code. This response code informs the worker what to do with the job after processing.

Step 4 - Execute response task A worker informs the dispatcher of the job status after completion based on the job response. If the MusicBrainz data request is successful (Worker n_2), a set of validation criteria is used. If validation is successful, Worker n_2 will inform the scheduler to prepare a Spotify-to-MusicBrainz database write into GRAIL.

Step 5 - Send status code to dispatcher Responses from workers in Step 4 are sent to the dispatch. The dispatcher processes these responses, and will prepare new jobs if the response requires it. Python script running periodically, accesses new database updates and generates new seed data to submit to the dispatch.

3. API ACCESS

GRAIL will become open for public use in the coming months with documentation from the endpoint: `api.digitalmusiclab.org`. Users can query the API using track, release, and artist IDs and names, from a documented list of services and datasets. Access keys will be granted through a free registration process available via a registration portal. API keys will have reasonable rate limits based on our current server restrictions. The terms of service will state that this API is for and by the Creative Commons, and is designed for use in music research. Users will not be able to use GRAIL for profit without direct permission of the services used in the application.

[3] Although we cannot be fully certain of the veracity of the Nokia Music-ISRC mapping, part of the intention of GRAIL is to test the accuracy of these links through cross-validation with multiple resources. ISRCs are used as a starting point and are not blindly trusted. Reliability comes from consistency of data across multiple, independent resources that may, or may not provide ISRC information.

[4] `https://developer.spotify.com/web-api/endpoint-reference/`

API ID Space	No. Linkages
Spotify track IDs	17.5M
Echo Nest track ID	6.5M
MusicBrainz track ID	465K
Million Song Dataset track ID	521K
MusixMatch track ID	3.8M
LyricFind track ID	1.5M
Spotify release ID	1.5M
MusicBrainz release ID	163K
MusicBrainz release Group ID	97K
Spotify artist ID	6.6M
MusicBrainz artist ID	203K
MusixMatch artist ID	2.1M
Jambase artist ID	860K
OpenAura artist ID	3.7M
SeatGeek artist ID	2.8M
Seatwave artist ID	740K
LyricFind artist ID	1.5M
Twitter artist ID	670K
Facebook artist ID	3.7M
Tumblr artist ID	27K
Free Music Archive artist ID	240K
7digital artist ID	5.5M

Table 1: Current number of linkages from various APIs into GRAIL.

4. CONCLUSIONS

GRAIL will hopefully become an important resource for the MIR community. This conviction is due to the fact that GRAIL is designed constantly to check and update its content over time. In sum, GRAIL i) publishes criteria information relating to each linkage, ii) clearly documents how IDs are prioritized when alternatives exist, iii) provides every possible ID rather than just one, and iv) timestamps all linkages to provide further contextual information, enabling researchers to make critical judgments with respect to data integrity. Our intention is that the database will inspire a degree of confidence, commensurate with high-level research.

5. REFERENCES

[1] B. Angeles, C. McKay, and I. Fujinaga. Discovering metadata inconsistencies. In *Proceedings of the 11th International Society for Music Information Retrieval*, pages 195–200, 2010.

[2] D. Baur, T. Langer, and A. Butz. Shades of music: Letting users discover sub-song similarities. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 111–116, 2009.

[3] S.J. Downie. The music information retrieval evaluation exchange (mirex). *D-Lib Magazine*, 12(12):795–825, 2006.

[4] J. Hemerly. Making metadata: The case of musicbrainz. *Available at SSRN 1982823*, 2011.

[5] C. McKay, J.A. Burgoyne, J. Hockman, J.B.L. Smith, G. Vigliensoni, and I. Fujinaga. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. *Proceedings of the 11th International Society for Music Information Retrieval*, 2010.

[6] F. Pachet. Musical metadata and knowledge management. In *Encyclopedia of Knowledge Management*, pages 672–677. 2006.

[7] A. Schindler and A. Rauber. Capturing the temporal domain in echo nest features for improved classification effectiveness. In *International Workshop on Adaptive Multimedia Retrieval*, pages 214–227, 2012.

[8] G. Vigliensoni, J.A. Burgoyne, and I. Fujinaga. Musicbrainz for the world: the chilean experience. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 131–136, 2013.