EVALUATING THE GENRE CLASSIFICATION PERFORMANCE OF LYRICAL FEATURES RELATIVE TO AUDIO, SYMBOLIC AND CULTURAL FEATURES

Cory McKay, John Ashley Burgoyne, Jason Hockman, Jordan B. L. Smith, Gabriel Vigliensoni and Ichiro Fujinaga
Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
McGill University, Montréal, Québec, Canada
[cory.mckay,jason.hockman,jordan.smith2]@mail.mcgill.ca,
[ashley,gabriel,ich]@music.mcgill.ca

ABSTRACT

This paper describes experimental research investigating the genre classification utility of combining features extracted from lyrical, audio, symbolic and cultural sources of musical information. It was found that cultural features consisting of information extracted from both web searches and mined listener tags were particularly effective, with the result that classification accuracies were achieved that compare favorably with the current state of the art of musical genre classification. It was also found that features extracted from lyrics were less effective than the other feature types. Finally, it was found that, with some exceptions, combining feature types does improve classification performance. The new lyricFetcher and jLyrics software are also presented as tools that can be used as a framework for developing more effective classification methodologies based on lyrics in the future.

1. INTRODUCTION

Automatic music classification is an important area of music information retrieval (MIR) research. Areas such as classification by genre, mood, artist and user tag have all received significant attention in the MIR literature. Classification is typically performed by training machine learning algorithms on features extracted from audio recordings, symbolic data or cultural information mined from the Internet. An interest in features extracted from textual transcriptions of lyrics has also become increasingly evident recently.

Most research to date has involved experiments involving one or, at most, two of these four types of data. This leaves unanswered questions as to whether improvements in classification performance might be achieved by combining features extracted from various combinations of these four musical data sources, especially with respect to the relatively new area of classification based on lyrics.

The first goal of the research presented here is to investigate this issue through a series of genre classification experiments on each possible subset combination of features extracted from lyrical, audio, symbolic and cultural data. Genre classification in particular is chosen because it is a well-established area of inquiry in the MIR literature that can be particularly difficult to perform well, and as such provides a good general basis for evaluation.

The second goal of this paper is to present software for mining lyrics from the Internet and for extracting features from them. There is not yet an established research toolset for performing these tasks, and the lyricFetcher and jLyrics software described here are intended to fill this gap.

2. PREVIOUS RESEARCH

2.1 Mining Lyrics from the Web

There are many web sites providing access to lyric transcriptions, including industry-approved pay services (e.g., Gracenote Lyrics), specialized lyric-scraping services (e.g., EvilLyrics, iWeb Scraping and Web Data Extraction), and other sites that amalgamate user contributions. The main difficulties encountered when automatically mining lyrics are associated with high variability in display formatting and content. Many sites also attempt to obscure lyrical content in the page source because of copyright concerns. There have been several attempts to extract and align lyrics from multiple sources automatically using dynamic programming [2,6], but these have encountered difficulties due to varying search results.

LyricsFly is one site that promises well-formatted lyrics and simplified searches accessible via a published API. Lyrics are provided in a convenient XML format, and multiple versions of songs are accessible. LyricWiki once provided a public API as well, but has since discontinued this service due to copyright concerns. Its content is still accessible via web browsing, however.

2.2 Extracting Classification Features from Lyrics

Logan et al. [10] and Mahedero et al. [11] provide important early contributions on analyzing lyrics using a

variety of techniques drawn from natural language processing, including topic modelling, to quantify musical similarity. Maxwell [12] also uses a large and varied feature set extracted from lyrics to rank similarity.

Mayer et al. [13] provide a particularly helpful examination of the classificatory power of various lyrical features with respect to genre. Kleedorfer et al. [5] and Wei et al. [18] present strategies for identifying topics, which can be adapted for use as classification features. Hirjee and Brown [3] present a sophisticated tool for extracting rhymes from lyrics, with a focus on hip-hop styles.

Some research has been performed on combining lyrical features with audio features in the context of artist, genre and mood classification [4,7,8,13]. Brochu and Freitas [1] have done research on combining lyrical features with features extracted from symbolic music.

2.3 jMIR

jMIR [14] is a suite of software tools and other resources developed for use in automatic music classification research. It was used to perform all of the experiments described in this paper. jMIR includes the following components:

- **jAudio:** An audio feature extractor.
- **jSymbolic:** A symbolic feature extractor.
- **jWebMiner 2.0:** A cultural feature extractor.
- **ACE 2.0:** A metalearning-based classifier.
- **jMusicMetaManager:** Software for managing and detecting errors in musical datasets.
- **jMIRUtilities:** Performs infrastructural tasks.
- **ACE XML:** Standardized MIR file formats.
- **Codaich, Bodhidharma MIDI and SAC:** Musical research datasets.

The jMIR software is all implemented in Java, which has advantages with respect to platform-independence. All jMIR components are open-source and are distributed free of charge at jmir.sourceforge.net.

2.4 Comparing the Performance of Feature Types

This paper expands upon the research described in [15], which experimentally investigated the classification utility of combining features extracted from audio, symbolic and cultural sources of musical information using an earlier version of jMIR. It was found that combining feature types did indeed substantially improve classification performance, in terms of both overall classification accuracy and the seriousness of those misclassifications that did occur.

To the best of the authors' knowledge, [15] is the only previous study involving cultural, symbolic and audio data. There have, however, been many important studies involving features extracted from pairs of musical data types, including [9] and [19]. Section 2.2 highlights additional work involving lyrics.

3. THE SLAC DATASET

The new SLAC (Symbolic Lyrical Audio Cultural) dataset is an expansion of the SAC dataset [15] that now includes lyrics. The purpose of this dataset is to facilitate experiments comparing the relative performance of features extracted from different types of musical data.

SAC consists of 250 MP3 recordings, 250 matching MIDI recordings and identifying metadata for each recording. This metadata is stored in an iTunes XML file that can be parsed by software such as jWebMiner in order to extract cultural features from the web.

SLAC adds lyrics to all of the non-instrumental musical pieces in SAC. These lyrics were mined from the Internet, as described in Section 4.

SLAC is divided into 10 genres, with 25 pieces of music per genre. These 10 genres consist of 5 pairs of similar genres, as shown in Figure 1. This arrangement makes it possible to perform 5-class genre classification experiments as well as 10-class experiments simply by combining each pair of related genres into one class, thus providing an indication of how well systems perform on both small and moderately sized genre taxonomies.

Blues: Modern Blues *and* Traditional Blues
Classical: Baroque *and* Romantic
Jazz: Bop *and* Swing
Rap: Hardcore Rap *and* Pop Rap
Rock: Alternative Rock *and* Metal

Figure 1: The ten genres found in the SLAC dataset and the five super-genres that they can be paired into.

SLAC includes some instrumental music. This complicates classification based on lyrics, as lyrics provide no way to distinguish one instrumental piece from another. Nonetheless, the inclusion of some instrumental music is necessary to evaluate classification performance properly, as one must simulate the music that classification systems will encounter in practice, including instrumental music.

4. MINING LYRICS WITH LYRICFETCHER

A new lyrics mining script called *lyricFetcher* was implemented in Ruby to automatically harvest lyrics from LyricWiki and LyricsFly. These two repositories were chosen for their large sizes and because of the simplicity of querying their collections: LyricsFly provides a simple API and LyricWiki offers a standardized URL naming scheme that is relatively easy to mine.

Once provided with a list of artist names and song titles to search for, lyricFetcher obtains lyrics in three steps: first, a query is made to the lyrics source; second, the lyrics themselves are extracted from the result; and third, lyrical content is cleaned and standardized in post-processing, an important step given the variability in for-

matting of user-contributed lyrics. In particular, raw retrieved lyrics are often abridged by providing a label for the first occurrence of a section (e.g., "chorus," "hook," "refrain," etc.) and repeating only this label when the section reoccurs. lyricFetcher automatically searches for and expands such sections. Common keywords added to the lyrical transcriptions, such as "verse," are also removed.

lyricFetcher was used to mine LyricWiki and LyricsFly for the lyrics to the recordings in Codaich and SLAC. These lyrics were used in the experiments described below in Section 6. Lyrics were manually retrieved from other web sources for the 20 pieces out of the 160 non-instrumental pieces in SLAC for which lyrics could not be harvested automatically from LyricWiki and LyricsFly.

5. EXTRACTING FEATURES FROM SLAC

5.1 Lyrical Features Extracted

A large number of features were implemented and extracted based on a survey of previous work and on original ideas: *AutomatedReadabilityIndex, AverageSyllablesPerWord, ContainsWords, FleshKincaidGradeLevel, FleshReadingEase, FunctionWordFrequencies, LetterBigramComponents, LetterFrequencies, LettersPerWordAverage, LettersPerWordVariance, LinesPerSegmentAverage,*[1] *LinesPerSegmentVariance, NumberOfLines, NumberOfSegments, NumberOfWords, PartOfSpeechFrequencies,*[2] *PunctuationFrequencies, RateOfMisspelling, SentenceCount, SentenceLengthAverage, TopicMembershipProbabilities,*[3] *VocabularyRichness, VocuabularySize, WordProfileMatch, WordsPerLineAverage* and *WordsPerLineVariance*. Descriptions of these features are provided at jmir.sourceforge.net/index_jLyrics.html.

5.2 The jLyrics Feature Extractor

A new Java-based feature extraction framework called *jLyrics* was implemented as part of this research. Like the existing jMIR feature extractors, it is designed to serve as an easy-to-use feature extraction application as well as an extensible framework for developing new features. It has the usual jMIR advantages in this respect [14], including a modular architecture, automatic resolution of feature dependencies and the option of saving feature values in several file formats. Many of the features described in Section 5.1 were implemented directly in jLyrics, although some features based on third-party libraries remain to be ported to the Java framework.

In addition to extracting features jLyrics can, given sets of lyrics belonging to a class, generate profiling reports indicating ranked lists of the most commonly used words in each class. These profiles can be used to "train" *WordProfileMatch* features to measure how well novel lyrics match each class' profile. Lyrics mined with lyricFetcher for the music in Codaich (with all pieces in SLAC filtered out) were used to do just this, in preparation for the experiments described in Section 6.

5.3 Audio, Symbolic and Cultural Feature Extraction

jMIR, as described in Section 2.3 and [14], was used to extract audio, symbolic and cultural features from SLAC. Of particular interest, the new jWebMiner 2.0 [17] software was used to extract cultural features based on both Yahoo! co-occurrence page counts and Last.FM user tags, as opposed to the older jWebMiner 1.0 used in [15], which only extracted features based on web searches. A newer version of ACE, ACE 2.0, was also used.

6. EXPERIMENTAL PROCEDURE

The first step of the experiment was to extract feature values from SLAC, as described in Section 5. This resulted in a set of 26 features (A) extracted from the audio version of each piece, 101 features (S) extracted from the MIDI version of each piece, 26 features (L) extracted from the lyrics for each piece and 20 features (C) extracted from the Internet based on the identifying metadata for each piece.[4] These four types of features were then grouped into all 15 possible subset combinations using jMIRUtilites. These feature groups are identified using the codes indicated in Table 1.

Feature Types	Identifying Code
Symbolic	S
Lyrical	L
Audio	A
Cultural	C
Symbolic + Lyrical	SL
Symbolic + Audio	SA
Symbolic + Cultural	SC
Lyrical + Audio	LA
Lyrical + Cultural	LC
Audio + Cultural	AC
Symbolic + Lyrical + Audio	SLA
Symbolic + Lyrical + Cultural	SLC
Symbolic + Audio + Cultural	SAC
Lyrical + Audio + Cultural	LAC
Symbolic + Lyrical + Audio + Cultural	SLAC

Table 1: The identifying codes for the feature type groups used in each of the experiments.

[1] A "segment" is a unit of text separated by line breaks.
[2] Extracted using the Stanford parts-of-speech tagger [18].
[3] Trained on Codaich (with SLAC instances filtered out) using latent Dirichlet allocation [8].
[4] The jMIR feature extractors are each capable of extracting more features than this, but were set to omit unpromising features in order to save processing time. Also, many of the features that were extracted are in fact feature vectors consisting of multiple values.

Features	5-Genre Accuracy (%)	10-Genre Accuracy (%)
S	85	66
L	69	43
A	84	68
C	100	86
SL	89	70
SA	95	74
SC	99	89
LA	88	66
LC	100	81
AC	100	85
SLA	93	77
SLC	99	84
SAC	100	89
LAC	99	83
SLAC	99	85

Table 2: Classification accuracies for each of the experiments. Feature codes are identified in Table 1. All values are averages across cross-validation folds.

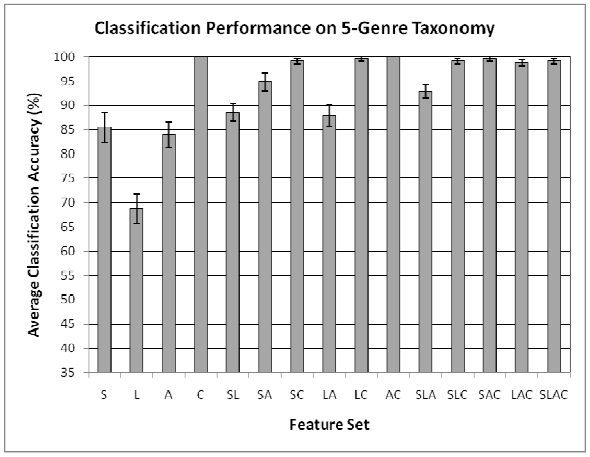

Figure 2: Results of the 5-genre experiments, as detailed in Table 2. Feature set codes are defined in Table 1.

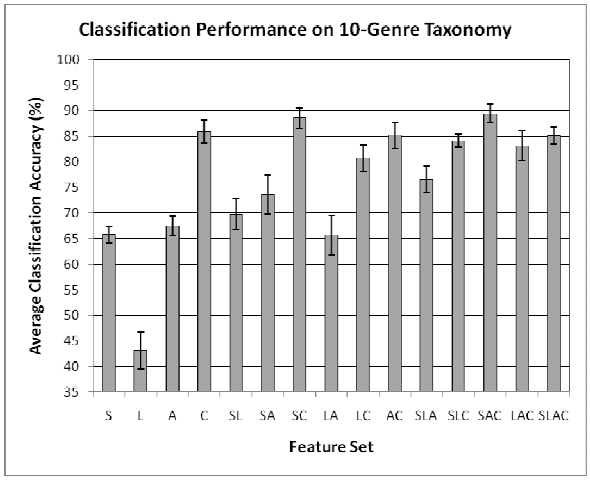

Figure 3: Results of the 10-genre experiments, as detailed in Table 2. Feature set codes are defined in Table 1.

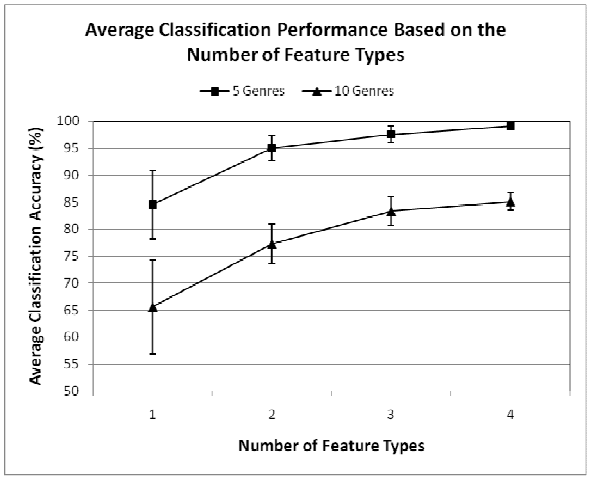

Figure 4: Classification accuracies averaged for all groups of, respectively, 1, 2, 3 and 4 feature types.

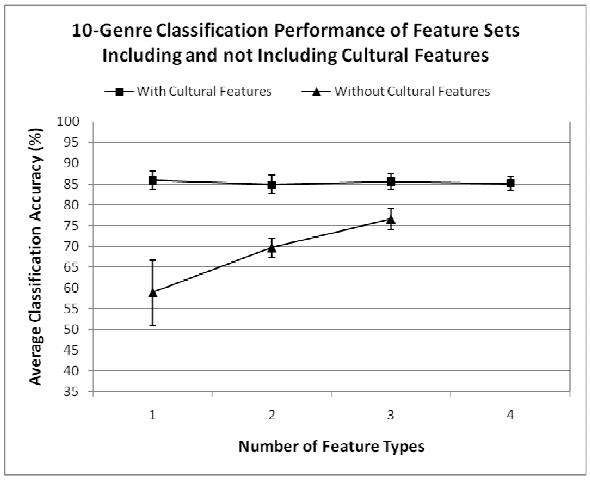

Figure 5: Average accuracies for feature groups including cultural features (C), compared to groups without C.

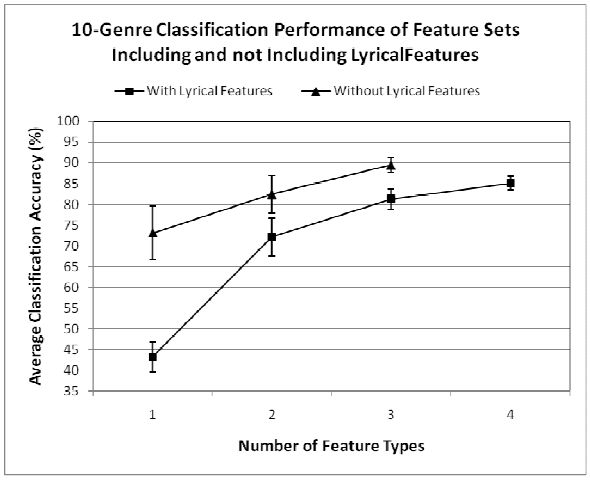

Figure 6: Average accuracies for feature groups including lyrical features (L), compared to groups without L.

jMIR ACE 2.0 was then used to classify each of these 15 feature sets by genre in 30 separate 10-fold cross-validation metalearning-based experiments, [5] such that each of the 15 feature sets was processed once in a 5-genre experiment and once in a 10-genre experiment. The results of these experiments are shown in Table 2. Figures 2 and 3 also represent this information graphically for the 5- and 10-genre taxonomies, respectively. The error bars on all figures represent standard error (i.e., the standard deviation of the cross-validation accuracies divided by the square root of the number of measurements).

7. RESULTS AND DISCUSSION

7.1 Overall Classification Performance

Overall, excellent classification accuracies were obtained with jMIR, with peak performances of 100% on the 5-genre taxonomy and 89% on the 10-genre taxonomy. For the purpose of comparison, the MIREX (www.music-ir.org/mirex/2010/) contests provide the best benchmarking references available. The highest MIREX symbolic genre classification performance to date is 84%, attained on a 9-genre ontology, and all six audio genre classification evaluations to date on genre ontologies larger than six classes have failed to achieve success rates above 80%. Although it is inappropriate to compare results obtained on different datasets directly, this does cast the results obtained here with jMIR in a favourable light.

7.2 Effect on Accuracy of Combining Feature Types

The next thing to consider was, now that lyrical features were included and the new jWebMiner 2.0 cultural features were used, whether combining different feature types still improved classification performance, as was the case in [15]. Figure 4 demonstrates the results of averaging together the classification accuracies of all feature groups with the same number of feature types (i.e., S, L, A and C; SL, SA, SC, LA, LC and AC; etc.), with a separate curve for each of the two genre taxonomies. It can be seen that, on average, classification accuracy did indeed increase with the number of feature types available.

It thus appears, at least upon first consideration, that combining features from different types of data does tend to improve performance. A closer examination of Table 2 shows that this was only true on average, however, as there were some cases where combining feature groups actually decreased performance (e.g., LC performed less well than C in the 10-genre experiments). Furthermore, an examination of Figure 5, described below, suggests that there was no advantage to combining cultural features in general with any other feature types.

7.3 Effectiveness of Cultural Features

Figure 5 shows, for the 10-class taxonomy, the average performance of all feature groups of the same size that contain cultural features, compared with the average performance of all feature groups of the same size that do not contain cultural features. The experimental results as a whole demonstrate that, for both taxonomies, cultural features significantly outperformed all other feature types.[6]

This dominance of cultural features was not apparent in [15], which only used cultural features derived from web searches. As described in [17], the new jWebMiner 2.0 combines these features with additional tag-based features extracted from Last.FM. This is likely responsible for the much higher performance of cultural features in this study relative to the results from [15].

7.4 Effectiveness of Lyrical Features

Figure 6 shows, for the 10-class taxonomy, the average performance of all feature groups of the same size that contain lyrical features, compared with the average performance of all feature groups of the same size that do not contain lyrical features. The results indicate that lyrical features were significantly less effective than the other feature types.[7] It is notable, however, that combining lyrical features with other feature types did, in some but not all cases, improve performance relative to the features operating individually. This is true for SL and SLA in both the 5- and 10-genre experiments. Furthermore, it is important to emphasize that 90 of the SLAC recordings were instrumental (although these recordings were strongly correlated with the Jazz and Classical genres).

8. CONCLUSIONS

This paper introduces the lyricFetcher and jLyrics tools for, respectively, mining lyrics from the web and extracting features from them. These tools are available for use in other research projects, and jLyrics in particular is designed to provide an easily extensible framework for implementing, testing and extracting new features.

With respect to the experiments described in this paper, excellent overall classification accuracies were obtained relative to the current state of the art of genre clas-

[5] A validation partition was reserved for each of the 30 experiments in order to guard against overfitting. Any experiment that resulted in an average cross-validation success rate that was higher than the validation performance with statistical significance was redone. It should also be noted that ACE includes dimensionality reduction functionality, so training was actually performed with automatically chosen subsets of the available features in order to avoid the "curse of dimensionality."

[6] Based on a Wilcoxon signed-rank test with a significance level of 0.05.

[7] Based on a Wilcoxon signed-rank test with a significance level of 0.05.

sification. In particular, the jWebMiner 2.0 cultural features based on both web searches and listener tags extracted from Last.FM were especially effective. It was also found that combining different feature types improved performance on average if cultural features were unavailable, but was not necessary if cultural features were available. With respect to lyrical features, it was found that combining them with other types of features did, in certain cases, improve classification performance. Overall, however, lyrical features performed poorly relative to the other feature types.

The disappointing performance of the lyrical features was probably due in part to noisiness in the mined lyrical transcriptions, including inconsistent annotation practices, occasional errors and the inclusion of non-standardized markup in XML and other formats. The relatively low performance of lyrics was likely also partly due to inherent limitations with respect to classifying instrumental music, as well as to the general-purpose text mining orientation of the lyrical features used. This highlights the need for continued research on more specialized music-oriented lyrical features, and on still better lyric mining and cleaning methodologies. Both of these could potentially lead to significantly improved performance by lyrical features.

9. REFERENCES

[1] Brochu, E., and N. de Freitas. 2003. "Name that song!": A probabilistic approach to querying music and text. In *Advances in Neural Information Processing Systems* 15, 1505–12. Cambridge, MA: MIT Press.

[2] Geleijnse, G., and J. Korst. 2006. Efficient lyrics extraction from the web. *Proc. of the Int. Conference on Music Information Retrieval*. 371–2.

[3] Hirjee, H., and D. G. Brown. 2009. Automatic detection of internal and imperfect rhymes in rap lyrics. *Proc. of the Int. Society for Music Information Retrieval Conference*. 711–6.

[4] Hu, X, J. S. Downie, and A. F. Ehman. 2009. Lyric text mining in music mood classification. *Proc. of the Int. Society for Music Information Retrieval Conference*. 411–6.

[5] Kleedorfer, F., P. Knees, and T. Pohle. 2008. Oh oh oh whoah! Towards automatic topic detection in song lyrics. *Proc. of the Int. Conference on Music Information Retrieval*. 287–92.

[6] Knees, P., M. Schedl, and G. Widmer. 2005. Multiple lyrics alignment: Automatic retrieval of song lyrics. *Proc. of the Int. Conference on Music Information Retrieval. 564–*9.

[7] Laurier, C., J. Grivolla, and P. Herrera. 2008. Multimodal music mood classification using audio and lyrics. *Proc. of the Int. Conference on Machine Learning and Applications.* 688–93.

[8] Li, T., and M. Ogihara. 2004. Semi-supervised learning from different information sources. *Knowledge and Information Systems* 7 (3): 289–309.

[9] Lidy, T., A. Rauber, A. Pertusa, and J. M. Iñesta. 2007. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. *Proc. of the Int. Conference on Music Information Retrieval*. 61–6.

[10] Logan, B., A. Kositsky, and P. Moreno. 2004. Semantic analysis of song lyrics. *Proc. of the IEEE Int. Conference on Multimedia and Expo*. 827–30.

[11] Mahedero, J. P. G., Á. Martínez, and P. Cano. 2005. Natural language processing of lyrics. *Proc. of the ACM Int. Conference on Multimedia.* 475–8.

[12] Maxwell, T. 2007 Exploring the Music Genre: Lyric Clustering with Heterogeneous Features. *M.Sc. thesis,* University of Edinburgh.

[13] Mayer, R., R. Neumayer, and A. Rauber. 2008. Combination of audio and lyrics features for genre classification in digital audio collections. *Proc. of the ACM Int. Conference on Multimedia*. 159–68.

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

[15] McKay, C., and I. Fujinaga. 2008. Combining features extracted from audio, symbolic and cultural Sources. *Proc. of the Int. Conference on Music Information Retrieval*. 597–602.

[16] Neumayer, R., and A. Rauber. 2007. Integration of text and audio features for genre classification in music information retrieval. *Proc. of the European Conference on IR Research*. 724–7.

[17] Vigliensoni, G., C. McKay, and I. Fujinaga. 2010. Using jWebMiner 2.0 to improve music classification performance by combining different types of features mined from the web. Accepted for publication at the *Int. Society for Music Information Retrieval Conference*.

[18] Wei, B., C. Zhang, and M. Ogihara. 2007. Keyword generation for lyrics. *Proc. of the Int. Conference on Music Information Retrieval*. 121–2.

[19] Whitman, B., and P. Smaragdis. 2002. Combining musical and cultural features for intelligent style detection. *Proc. of the Int. Symposium on Music Information Retrieval*. 47–52.