SEARCH & SELECT – INTUITIVELY RETRIEVING MUSIC FROM LARGE COLLECTIONS

Peter Knees
Department of Computational Perception
Johannes Kepler University Linz, Austria

ABSTRACT

A retrieval system for large-scale collections that allows users to search for music using natural language queries and relevance feedback is presented. In contrast to existing music search engines that are either restricted to manually annotated meta-data or based on a query-by-example variant, the presented approach describes audio pieces via a traditional term vector model and allows therefore to retrieve relevant music pieces by issuing simple free-form text queries. Term vector descriptors for music pieces are derived by applying Web-based and audio-based similarity measures. Additionally, as the user selects music pieces that he/she likes, the subsequent results are adapted to accommodate to the user's preferences. Real-world performance of the system is indicated by a small user study.

1 MOTIVATION AND CONTEXT

The recent achievements of the MIR community have led to many innovative and possibly unconventional approaches to support users in finding desired music. Frequently, content-based analysis of the audio files or collaborative recommendations to point users to music they might like (e.g. [2]) are applied. Some approaches also incorporate information from different sources to build interactive interfaces (e.g. [6]).

However, to retrieve music from large databases, many approaches rely on *query-by-example* methods where the query must consist of a piece of information that has a representation similar to the records in the database, e.g. in *query-by-humming/singing* systems. Since users are accustomed to text-based search engines, which have become the "natural" way to find and access all other types of content like images, videos, and text, query-by-example music search systems often lack broad acceptance. On the other hand, systems that offer textual queries, e.g. catalog search engines of commercial music re-sellers, permit only filtering based on attributes like artist, album, track name, year, or subjective labels like genre or style.

To address some of these limitations, in [1], a system is presented that relies on a semantic ontology containing relations between meta-data and automatically extracted acoustic properties and that can be queried via natural language phrases. In the retrieval process, textual queries must then be mapped to the semantic concepts. With this system, semantic queries like *"something fast from..."* or *"something new from..."* can be processed. However, as with the traditional systems, the *cultural context* of the indexed music pieces is ignored.

For real-world applicability of a music search engine, it must in fact behave like a Web search engine like Google or Yahoo! and allow arbitrary queries like *rock with great riffs* or even *melodic metal with opera singer as front woman*. Our first steps into this direction can be found in [3]. By using Web-based features to describe music pieces in a collection, each piece can be represented by a term vector. Furthermore, audio-based similarity is incorporated to describe also those pieces for which no information can be found on the Web. Thus, we combine information about the *context* with information about the *content*. For retrieval, queries are sent to Google and the resulting Web pages are used to construct a query vector which can be compared to the term vectors of the pieces in the collection. In [4], we have modified this approach to include relevance feedback and to use a local Web page index for query vector construction instead of Google.

In this work, the applicability of the proposed methods is demonstrated. The user can simply initiate the search for desired music by typing some descriptive terms. From the returned items, those after the user's fancy are selected and transferred into a list of "harvested music pieces" (analogous to e.g. a shopping cart in an on-line shop). Based on the chosen music pieces, the consecutively presented results are modified such that they tend to contain more pieces similar to the ones in the "harvest list". The user can continue searching by selecting (or ignoring) more results or by issuing the next query.

2 TECHNICAL FUNDAMENTALS

In this section, we briefly review the technical basis of the presented application. Details can be found in [3, 4].

We derive track specific information from the Web by combining the results of three queries issued to Google:

1. "*artist*" music
2. "*artist*" "*album*" music review
3. "*artist*" "*title*" music review -lyrics

For each query, at most 100 of the top-ranked Web pages are retrieved and joined into a single set. All retrieved pages are cleaned from HTML tags and stop words in six languages. From each track's set, a modified $tf \times idf$ representation is calculated.

To complement context-based features with information on the content of the music, Single Gaussian MFCC distribution models are calculated for each track [5]. Since the Kullback-Leibler divergence, which is applied for similarity computation of the models, has some undesirable consequences, a rank-based correction called *Proximity Verification* is used for post-processing [7].

The usage of track features from two distinct sources can be used *(a)* to reduce the dimensionality of the term vector space (via a modified χ^2 test), *(b)* to emphasize terms that occur frequently among similar sounding pieces, and – most important – *(c)* to describe music pieces with no (or few) associated information present on the Web. The last two points are achieved by performing a Gaussian weighting over the 10 acoustically nearest neighbors' term vectors.

Given a query, a vector space representation is constructed of the 20 most relevant Web pages from the set of retrieved pages, i.e. the pages used to construct the track-specific term vectors. Using the obtained query vector, distances to all tracks in the collection are calculated and ranked. The top results are then returned to the user. Selection of tracks marks them as relevant and results in a re-weighting of the query vector toward the relevant and away from the non-relevant pieces. This is accomplished by incorporating Rocchio's relevance feedback method [8].

3 USER STUDY

A small user study with 11 participants was conducted to get an impression of the retrieval application's performance. To this end, each participant was asked to submit 5 queries of choice to the system. Thus, in total, 55 different queries have been issued by the participants. For each query, 100 results were presented in groups of 20; selection of tracks influenced the next 20 results. From the set of issued queries, 6 basic types of queries could be identified. Since searching for lyrics is currently not supported, queries addressing lyrics are not included.

4 RESULTS AND CONCLUSIONS

Table 1 shows the different categories as well as the number of queries belonging to these categories. Note that a query can be assigned to multiple categories (e.g. *vienna electro dj* or *rammstein music with strong keyboard*). Worst results can be observed for queries that are aimed to retrieve a specific track. Although the user may select tracks other than the one specified, naturally the number is very low. Furthermore, it can be seen that users are most satisfied with results for genre queries (e.g. *eurodance*) and geographically related queries (e.g. *new orleans*).

category	*# queries*	*avg. relevant*
genre	28	33.25
artist	12	28.50
instrumentation	7	24.71
track	6	2.50
geographical	5	33.00
movie related	4	16.00
other	3	23.33
total	*55*	*29.75*

Table 1. Identified query categories, the number of queries belonging to these categories, and the average number of relevant music pieces (out of 100).

5 ACKNOWLEDGMENTS

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF project number L112-N04) and the Vienna Science and Technology Fund (WWTF project CIO10 "Interfaces to Music").

6 REFERENCES

[1] S. Baumann, A. Klüter, and M. Norlien. Using natural language input and audio analysis for a human-oriented MIR system. In *Proc. 2nd WEDELMUSIC*, Darmstadt, Germany, 2002.

[2] O. Celma, M. Ramirez, and P. Herrera. Foafing the music: A music recommendation system based on RSS feeds and user preferences. In *Proc. 6th ISMIR*, London, UK, 2005.

[3] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A Music Search Engine Built upon Audio-based and Web-based Similarity Measures. In *Proc. 30th ACM SIGIR*, Amsterdam, the Netherlands, 2007.

[4] P. Knees and G. Widmer. Searching for Music Using Natural Language Queries and Relevance Feedback. In *Proc. 5th Int'l Workshop on Adaptive Multimedia Retrieval*, Paris, France, 2007.

[5] M. Mandel and D. Ellis. Song-Level Features and Support Vector Machines for Music Classification. In *Proc. 6th ISMIR*, London, UK, 2005.

[6] E. Pampalk and M. Goto. Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. In *Proc. 7th ISMIR*, Victoria, Canada, 2006.

[7] T. Pohle, P. Knees, M. Schedl, and G. Widmer. Automatically Adapting the Structure of Audio Similarity Spaces. In *Proc. 1st Workshop on Learning the Semantics of Audio Signals*, Athens, Greece, 2006.

[8] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall, 1971.