A Document-centered Approach to a Natural Language Music Search Engine

Peter Knees, Tim Pohle, Markus Schedl,
Dominik Schnitzer, and Klaus Seyerlehner

Dept. of Computational Perception, Johannes Kepler University Linz, Austria
peter.knees@jku.at

Abstract. We propose a new approach to a music search engine that can be accessed via natural language queries. As with existing approaches, we try to gather as much contextual information as possible for individual pieces in a (possibly large) music collection by means of Web retrieval. While existing approaches use this textual information to construct representations of music pieces in a vector space model, in this paper, we propose a document-centered technique to retrieve music pieces relevant to arbitrary natural language queries. This technique improves the quality of the resulting document rankings substantially. We report on the current state of the research and discuss current limitations, as well as possible directions to overcome them.

1 Motivation and Context

While digital music databases contain several millions of audio pieces nowadays, indexing of these collections is in general still accomplished using a limited set of traditional meta-data descriptors like artist name, track name, album, or year. In most cases, also some sort of classification into coarse genres or different styles is available. Since this may not be sufficient for intuitive retrieval, several innovative (content-based) approaches to access music collections have been presented in the past years. However, the majority of these retrieval systems is based on *query-by-example* methods, i.e. the user must enter a query in a musical representation which is uncommon to most users and thus lacks acceptance. To address this issue, recently, different approaches to music search engines that can be accessed via textual queries have been proposed [4–6, 9].

In [6], we presented an approach that exploits contextual information related to the music pieces in a collection. To this end, *tf* × *idf* features are extracted from Web pages associated with the pieces and their corresponding artist. Furthermore, to represent audio pieces with no (or only little) Web information associated, also audio similarity is incorporated. This technique enables the user to issue queries like *"rock with great riffs"* to express the intention to find pieces that contain energetic guitar phrases instead of just finding tracks that have been labeled as *rock* by some authority. The general intention of the system presented in [6] is to allow for virtually any possible query and return the most appropriate pieces according to their "Web context" (comparable to e.g. Google's image search function).

In this paper, we present an alternative method to obtain a relevance ranking of music pieces wrt. a given query. Instead of constructing vector space representations for each music piece, we apply a traditional document indexing approach to the set of retrieved music-related Web pages and introduce a simple ranking function which improves the overall retrieval performance substantially.

2 Technical Background

Prior to presenting the modified retrieval approach, we briefly review the vector space-based method described in [6]. The first data acquisition step is identical for both approaches.

2.1 Vector Space Model approach (VSM)

To obtain as much track specific information as possible while preserving a high number of Web pages, for each track in the collection, three queries are issued to Google (at most 100 of the top-ranked Web pages are retrieved per query and joined into a single set):

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

After HTML tag and stop word removal, for each piece, all associated documents are treated as one large document and a weighted term vector representation is calculated using a modification of the $tf \times idf$ function.

In addition to the context-based features, information on the (timbral) content of the music is derived by calculating a Single Gaussian MFCC (*Mel Frequency Cepstral Coefficients*) distribution model for each track. Acoustic similarity between two pieces can be assessed by computing the Kullback-Leibler divergence on their models [8]. Based on the audio similarity information, feature space pruning is performed by applying a modified χ^2 test that simulates a 2-class discrimination task between the most similar sounding and the most dissimilar sounding tracks for each piece. For the evaluation collection used in [6], this step reduces the feature space from about 78,000 dimensions to about 4,700. Beside feature space reduction, the audio similarity measure can also be used to emphasize terms that occur frequently among similar sounding pieces, and – most important – to describe music pieces with no (or few) associated information present on the Web. These two tasks are achieved by performing a Gaussian weighting over the 10 acoustically nearest neighbors' term vectors.

After obtaining a term weight vector for each track in the music collection, natural language queries to the system are processed by adding the constraint *music* to the query and sending it to Google. From the 10 top-ranked Web pages, a query vector is constructed in the feature space. This query vector can then be compared to the music pieces in the collection by calculating cosine distances. Based on the distances, a *relevance ranking* is obtained.

2.2 Rank-based Relevance Scoring (RRS)

In contrast to the VSM method that relies on the availability of Google to process queries, we propose to directly utilize the Web content that has been retrieved in the data acquisition step. To this end, we create an off-line index of all pages using the open source package *Lucene* [1]. The usage of an off-line index allows to apply an alternative relevance ranking method since all indexed documents are at least relevant to one of the music pieces in the archive. Thus, we can take advantage of this information by exploiting these relations. More precisely, when querying the *Lucene* off-line index, a relevance ranking of the indexed documents according to the query is returned. Since we know for which music pieces these documents have been retrieved (and are thus relevant), we can simply create a set of music pieces relevant to the query by gathering all music pieces that are associated with at least one of the returned documents. Moreover, we can exploit the *ranking* information of the returned documents to introduce a very simple (but effective) relevance scoring function. Hence, for a given query q, we calculate the *rank-based relevance scoring* (RRS) for each music piece m as

$$RRS(m,q) = \sum_{p \in D_m \cap D_q} 1 + |D_q| - rank(p, D_q), \tag{1}$$

where D_m is the set of text documents associated with music piece m, D_q the set of relevant text documents with respect to query q, and $rank(p, D_q)$ a function that returns the rank of document p in the (ordered) set D_q (highest relevance corresponds to rank 1, lowest to rank $|D_q|$). Finally, the relevance ranking is obtained by sorting the music pieces according to their RRS value.

3 Evaluation and Discussion

To examine the impact of RRS on the retrieval quality, we have conducted various experiments on the same test collection as used in [6] for reasons of comparability. This collection consists of 12,601 unique tracks by 1,200 artists labeled with 227 different tags from Audioscrobbler/Last.fm [2, 3]. To measure retrieval performance, each of the 227 tags serves as query – music pieces are considered relevant iff they have been labeled with the corresponding tag. Examples for tags (and thus queries) are *hard rock*, *disco*, *soul*, *melancholy*, or *nice elevator music*. More details on the properties of the test collection can be found in [6].

Figure 1 depicts the *precision at 11 standard recall values* curves for RRS, the best scoring VSM approach from [6], and the baseline (giving indication of the "hardness" of the evaluation collection). At the (theoretical) 0.0 recall level, precision is at 0.75, which is 0.13 above the VSM approach. At the 0.1 recall level, the difference is even more evident: while the term vector approach yields around 0.37 in precision, RRS reaches a precision value of 0.66. Similar observations can be made for other IR measures, cf. Table 1. As can be seen, for the VSM approach, on average, five out of the first ten pieces are relevant, using the RRS ranking, seven out of ten pieces are relevant in average.

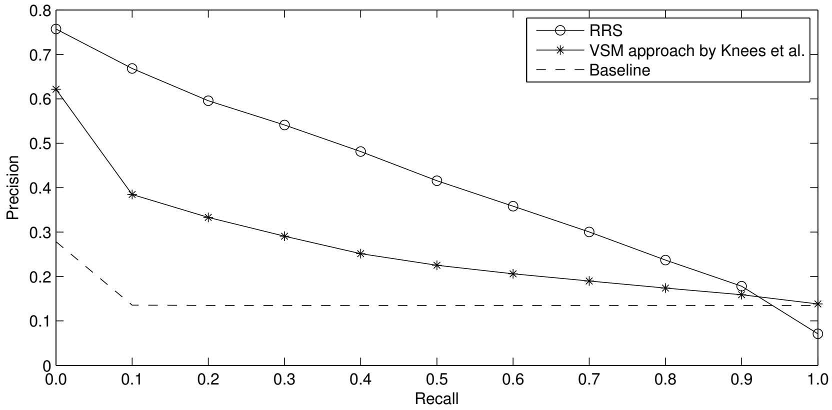

Fig. 1. Comparison of precision at 11 standard recall levels (avg. over all queries).

A major difference to the ranking functions based on term vector comparison is that the returned set of music pieces is in most cases only a subset of the whole collection. While term vector approaches always result in an ordering of the complete collection, using RRS returns only pieces that are assumed to be relevant somehow (by ignoring pieces with RRS scores of zero). For the user, it can be valuable information to know the total number of relevant pieces in the collection during examination of results. Furthermore, using the RRS scheme, it is not necessary to construct term vector representations for music pieces, which makes extraction of features as well as feature space pruning obsolete.

4 Conclusions and Future Work

We presented an alternative ranking approach for a natural language music search engine. By creating an off-line Web page index to process arbitrary queries and incorporating a simple ranking method, we were able to improve retrieval quality considerably. A possible explanation for the outcome that a rank-counting method outperforms an ordinary vector space model approach is

Ranking method	VSM	RRS
Average precision at seen relevant documents	25.29	**45.70**
R-Precision	26.41	**42.30**
Precision after 10 documents	49.56	**71.59**
Precision	13.47	**15.73**
Recall	**100.00**	87.81

Table 1. Selected single value summaries averaged over all queries (values in percent).

that important and diverse information contained on the Web pages may be improperly represented by a single term vector created from a merging of documents from different sources. In any case, the reasons for this finding have to be investigated more thoroughly in future work.

A current drawback of the proposed retrieval approach is the fact that it is not applicable to music pieces for which no associated information could be discovered via Web retrieval. As in [6], a possible solution could be to incorporate the audio similarity information, e.g., to associate Web pages related to a music piece also to similar sounding pieces. Furthermore, currently, possible queries to the system are limited by the vocabulary present on the retrieved pages. Future extensions could comprise a method that – again – sends a request to Google for queries containing unknown terms and, for example, finds those pages in the off-line index that are most similar to the top results from Google. From those pages, an RRS-based ranking could be obtained.

As for future work, we will also elaborate methods to incorporate relevance feedback into this new ranking mechanism, like it has already been successfully accomplished for the approach relying on term vector representations [7]. One possibility could be to propagate the feedback information back to the associated documents and perform relevance feedback on the document level. This could also allow for techniques like automatic query expansion. Finally, it is our goal to use a modified focused crawler that is specialized in indexing music related Web pages. Having a search index on our own would improve the applicability of the system and break the dependency on external Web search engines.

5 Acknowledgments

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L112-N04.

References

1. Apache Lucene. `http://lucene.apache.org`
2. Audioscrobbler. `http://www.audioscrobbler.net`
3. Last.fm. `http://www.last.fm`
4. S. Baumann, A. Klüter, and M. Norlien. Using natural language input and audio analysis for a human-oriented MIR system. In *Proc. 2nd WEDELMUSIC*, 2002.
5. O. Celma, P. Cano, and P. Herrera. Search Sounds: An audio crawler focused on weblogs. In *Proc. 7th ISMIR*, 2006.
6. 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*, 2007.
7. P. Knees and G. Widmer. Searching for Music Using Natural Language Queries and Relevance Feedback. In *Proc. 5th AMR*, 2007.
8. M. Mandel and D. Ellis. Song-Level Features and Support Vector Machines for Music Classification. In *Proc. 6th ISMIR*, 2005.
9. D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards Musical Query-by-Semantic-Description using the CAL500 Data Set. In *Proc. 30th ACM SIGIR*, 2007.