AUGMENTING TEXT-BASED MUSIC RETRIEVAL WITH AUDIO SIMILARITY

P. Knees[1], T. Pohle[1], M. Schedl[1], D. Schnitzer[1,2], K. Seyerlehner[1], and G. Widmer[1,2]
[1]Department of Computational Perception, Johannes Kepler University Linz, Austria
[2]Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria

ABSTRACT

We investigate an approach to a music search engine that indexes music pieces based on related Web documents. This allows for searching for relevant music pieces by issuing descriptive textual queries. In this paper, we examine the effects of incorporating audio-based similarity into the text-based ranking process – either by directly modifying the retrieval process or by performing post-hoc audio-based re-ranking of the search results. The aim of this combination is to improve ranking quality by including relevant tracks that are left out by text-based retrieval approaches. Our evaluations show overall improvements but also expose limitations of these unsupervised approaches to combining sources. Evaluations are carried out on two collections, one large real-world collection containing about 35,000 tracks and on the CAL500 set.

1. MOTIVATION AND RELATED WORK

In the last years, the development of *query-by-description* music search engines has drawn increasing attention [1–5]. Given the size of (commercial) digital music collections nowadays (several millions of tracks), this is not a big surprise. While most “traditional” music retrieval approaches pursue a *query-by-example* strategy, i.e., given a music piece, find me other pieces that sound alike, query-by-description systems are capable of retrieving relevant pieces by allowing to type in textual queries targeting musical or contextual properties beyond common meta-data descriptors. As this method of issuing queries is the common way to search the Web, it appears desirable to offer this type of functionality also in the music domain.

Several approaches to accomplish this goal have been presented – all of them with a slightly different focus. In [1], Baumann et al. present a system that incorporates meta-data, lyrics, and acoustic properties all linked together by a semantic ontology. Queries are analyzed by means of natural language processing and tokens have to be mapped to the corresponding concepts. Celma et al. [2] use a Web crawler focused on audio blogs and exploit the texts from the blogs to index the associated music pieces. Based on the text-based retrieval result, also musically similar songs can be discovered. In [3], we propose to combine audio similarity and textual content from Web documents obtained via Google queries to create representations of music pieces in a term vector space. A modification to this approach is presented in [6]. Instead of constructing term vector representations, an index of all downloaded Web documents is created. Relevance wrt. a given query is assessed by querying the Web document index and applying a technique called *rank-based relevance scoring* that takes into account the associations between music tracks and Web documents (cf. Section 2.1). Evaluations show that this document-centered approach is superior to the vector space approach. However, as this method is solely based on texts from the Web it may neglect important acoustic properties and suffer from effects such as popularity bias. Furthermore, inadequately represented tracks and tracks not present on the Web are penalized by this approach. In this paper, we aim at remedying these shortcomings and improving ranking quality by incorporating audio similarity into the retrieval process.

Recently, the method of relevance scoring has also been adapted to serve as a source of information for automatically tagging music pieces with semantic labels. In [5], Barrington et al. successfully combine audio content features (MFCC and Chroma) with social context features (Web documents and last.fm tags) via machine learning methods and therefore improve prediction accuracy. The usefulness of audio similarity for automatic tagging is also shown in [4] where tags from well-tagged tracks are propagated to untagged tracks based on acoustic similarity.

The remainder of this paper is organized as follows: In the next section we review methods for Web-based music track indexing and audio-based similarity computation. Section 3 describes two possible modifications of the initial approach that are examined in Section 4. In Section 5, based on these results, we discuss perspectives and limitations of combining Web- and audio-based approaches before drawing conclusions in Section 6.

2. INCORPORATED TECHNIQUES

In the following, we explain the methods for constructing a Web-based retrieval system and calculating audio similarity, which we combine in Section 3.

2.1 Web-based Indexing and RRS Ranking

The idea of Web-based indexing is to collect a high number of texts related to the pieces in the music collection to gather many diverse descriptions (and hence a rich indexing vocabulary) and allow for a large number of possible queries. In our first approach, we aimed at permitting virtually any query by involving Google for query expansion [3]. When introducing *rank-based relevance scoring (RRS)*, we renounced this step in favor of reduced complexity and improved ranking results [6]. From our point it is very reasonable to limit the indexing vocabulary to terms that actually co-occur with the music pieces (which is still very large). Construction of an index with a corresponding retrieval scheme is performed as follows.

To obtain a broad basis of track specific texts, for each music piece m in the collection M, three queries are issued to Google based on the information found in the id3 tags of the music pieces:

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

For each query, at most 100 of the top-ranked Web pages are retrieved and joined into a set (denoted as D_m in the following). For retrieval, we utilize the open source package *Nutch*[1]. Beside efficient retrieval, a further benefit is that all retrieved pages are also automatically indexed by *Lucene*[2] that uses a *tf*x*idf* variant as scoring function [7]. The resulting Web page index is then used to obtain a relevance ranking of Web pages for arbitrary queries. This page ranking, together with the information on associations between pages and tracks, serves as input to the RRS scheme. Compared to the original formulation in [6], we introduce the additional parameter n that is used to limit the number of top-ranked documents when querying the page index. For large collections, this is necessary to keep response time of the system short. For a given query q and for each music piece m, scores are calculated as:

$$RRS_n(m,q) = \sum_{p \in D_m \cap D_{q,n}} 1 + |D_{q,n}| - rnk(p, D_{q,n}), \tag{1}$$

where D_m is the set of text documents associated with music piece m (see above), $D_{q,n}$ the ordered set (i.e., the ranking) of n most relevant text documents with respect to query q, and $rnk(p, D_{q,n})$ a function that returns the rank of document p in $D_{q,n}$ (highest relevance corresponds to rank 1, lowest to rank $|D_{q,n}|$). The final relevance ranking of music tracks is then obtained by sorting the music pieces according to their RRS value.

Note that, as suggested in [5, 8], we also experimented with a weight-based version of relevance scoring (WRS) that incorporates the scores of the Web page retrieval step rather than the ranks. In our framework this modification worsened performance. Possible explanations are the differences in the underlying page scoring function or the different sources of Web pages (cf. [8]).

2.1.1 *Pseudo-Document Indexing*

Instead of modifying the page scoring scheme, we invented a simple alternative approach for text-based indexing that lies conceptually between the first vector space approach [3] and the relevance scoring scheme. For each music piece m, we concatenate all retrieved texts (i.e., all texts from D_m) into a single document which we index with *Lucene*. Hence, each music piece is represented by a single document that contains all relevant texts. Querying this *pseudo-document index* results directly in a ranking of music pieces. This rather "quick-and-dirty" indexing method will serve as a reference point in the evaluations and give insights into the capabilities of purely Web-based retrieval.

2.2 Audio-Based Similarity

For calculating music similarities, or more precisely, distances of tracks based on the audio content, we apply our algorithm which competed successfully in the "Audio Music Similarity and Retrieval" task of MIREX 2007 [9]. For each piece of music, *Mel Frequency Cepstral Coefficients (MFCCs)* are computed on short-time audio frames to characterize the frequency distribution of each frame and hence model aspects of timbre. On each frame, 25 MFCCs are computed. Each song is then represented as a *Gaussian Mixture Model (GMM)* of the distribution of MFCCs using a Single Gaussian Model with full covariance matrix [10]. The distance between these models is denoted by d_G.

Beside the MFCC-based distance component, also *Fluctuation Patterns (FPs)* are computed as proposed in [11]. A track is represented as a 12-band spectrogram and for each band, a *Fast Fourier Transformation (FFT)* of the amplitude is taken over a window of six seconds. The resulting matrix is referred to as the Fluctuation Pattern of the song. The FPs of two songs are compared by calculating the cosine distance, denoted by d_{FP}. Furthermore, two additional FP-related features are computed: *Bass (FPB)* and *Gravity (FPG)*. These two features are scalar and the distance between two songs is calculated by subtracting them, denoted by d_{FPB} and d_{FPG}. To obtain an overall distance value d measuring the (dis)similarity of two songs, all described distance measures are z-normalized and then combined by a simple arithmetic weighting:

$$d = 0.7 \cdot z_G + 0.1 \cdot (z_{FP} + z_{FPB} + z_{FPG}) \tag{2}$$

where z_x is the value of d_x after z-normalization. Finally, distances between two songs are symmetrized. For similarity computation, we ignore all pairs of songs by the same artist (artist filtering, cf. [12]) since this similarity is already represented within the Web features.

3. COMBINATION APPROACHES

This section describes two different approaches for combining the purely text-based retrieval approach with the

[1] http://lucene.apache.org/nutch

[2] http://lucene.apache.org

audio-based similarity information. According to [5, 13], the first approach can be considered an *early fusion* approach, since it incorporates the audio similarity information directly into the relevance scoring scheme, whereas the second approach can be considered a *late fusion* approach, since it modifies the ranking results obtained from the Web-based retrieval. Basically, both algorithms incorporate the idea of including tracks that sound similar to tracks already present through text-only retrieval. The score of a track m is calculated by summing up a score for being present in the text-based ranking and scores for being present within the nearest audio neighbors of tracks associated with the text-based ranking.

3.1 Modifying the Scoring Scheme (aRRS)

With this approach, we try to incorporate the audio similarity directly into the scoring scheme of RRS. The advantage is that this has to be calculated only once and does not require post-processing steps. The *audio-influenced RRS (aRRS)* is calculated as:

$$aRRS_n(m,q) = \sum_{p \in P_{m,q,n}} RF(p, D_{q,n}) \cdot MF(m,p), \quad (3)$$

$$RF(p, D_{q,n}) = 1 + |D_{q,n}| - rnk(p, D_{q,n}), \quad (4)$$

$$MF(m,p) = \alpha \cdot I(p, D_m) + \sum_{a \in A_m} I(p, D_a), \quad (5)$$

where $P_{m,q,n} = (D_m \cup D_{A_m}) \cap D_{q,n}$, $N_{a,k}$ the k nearest audio neighbors of a, A_m the set of all tracks a that contain m in their nearest audio neighbor set, i.e., all a for which $m \in N_{a,k}$, D_{A_m} the set of all documents associated with any member of A_m, and $I(x, D)$ a function that returns 1 iff $x \in D$ and 0 otherwise. Informally speaking, also tracks sounding similar to track m participate if a page relevant to m occurs in the page ranking for query q. The parameter α is used to control the influence of tracks that are directly associated with a Web page (in contrast to tracks associated via audio neighbors). In our experiments we set $\alpha = 10$. Note that aRRS is a generalization of RRS, as they are identical for $k = 0$.

3.2 Post-Hoc Audio-Based Re-Ranking (PAR)

The second approach incorporates audio similarity into an already existing ranking R. The advantage of this approach is that it can deal with outputs from arbitrary ranking algorithms. The post-hoc audio-based re-ranking (PAR) is calculated as:

$$PAR(m,R) = \sum_{t \in (m \cup A_m) \cap R} RF(t,R) \cdot NF(m,t), \quad (6)$$

$$NF(m,t) = \alpha \cdot I(m, \{t\}) + G(rnk(m, N_{t,k})) \cdot I(m, N_{t,k}), \quad (7)$$

$$G(i) = e^{-\frac{(i/2)^2}{2}} / \sqrt{2\pi}, \quad (8)$$

We included the gaussian weighting G in this re-ranking scheme because it yielded best results when exploring possible weightings. Parameter α can be used to control the scoring of tracks already present in R. Note that for $k = 0$, R remains unchanged.

4. EVALUATION

For evaluation, we decided to use two test collections with different characteristics. The first collection is a large real-world collection and contains mostly popular pieces. The second collection is the CAL500 set, a manually annotated corpus of 500 tracks by 500 distinct artists [14]. In the following, we describe both test collections in more detail.

4.1 The c35k Collection

The *c35k* collection is a large real-world collection, originating from a subset of a digital music retailer's catalog. The full evaluation collection contains about 60,000 tracks. Filtering of duplicates (including remixes, live versions, etc.; cf. [3]) reduces the number of tracks to about 48,000. As groundtruth for this collection, we utilize last.fm tags. Tags can be used directly as test queries to the system and serve also as relevance indicator (i.e., a track is considered to be relevant for query q if it has been tagged with tag q). From the 48,000 tracks, we were able to find track-specific last.fm tags for about 35,000 of the tracks. To obtain a set of test queries, we started with last.fm's list of top-tags and manually removed tags useless for our purpose (such as *seen live* or tags starting with *favorite*). We also searched for redundant tags (such as *hiphop*, *hip hop*, and *hip-hop*) and harmonized their sets of tagged tracks. However, all forms are kept as queries if they translate to different queries (in the example above, *hiphop* translates to a query with one token, *hip hop* to two tokens, and *hip-hop* to a phrase). As result, a set of 223 queries remained. From the 223 tags we further removed all tags with a number of associated tracks above the 0.95-percentile and below the 0.05-percentile, resulting in 200 test queries. A common way to increase the number of tagged examples is to use artist-specific tags if no track-specific tags are present [3, 8]. Since, in our indexing approach, tracks by the same artist share a large portion of relevant Websites, we decided against combination with artist tags to avoid overestimation of performance.

4.2 The CAL500 Set

The CAL500 set is a highly valuable collection for music information retrieval tasks [14]. It contains 500 songs (each from a different artist) which are manually annotated by at least three reviewers. Annotations are made wrt. a vocabulary consisting of 174 tags describing musically relevant concepts such as genres, emotions, acoustic qualities, instruments, or usage scenarios. Although we consider the fact that our indexing approach is in principle capable of dealing with large and varying vocabularies, some of these tags are not directly suited as query, especially negating concepts (e.g., *NOT-Emotion-Angry*) can

	Recall			**Precision**			**Prec@10**			**r-Precision**			**Avg. Prec. (MAP)**		
Baseline	100.00			3.65			3.60			3.65			3.68		
	Web only		PAR	Web only		PAR	Web only		PAR	Web only		PAR	Web only		PAR
PseudoDoc	93.66		**98.79**	**4.27**		3.67	**39.25**		17.40	**30.78**		22.94	**25.97**		18.81
	RRS	aRRS	PAR	RRS	aRRS	PAR	RRS	aRRS	PAR	RRS	aRRS	PAR	RRS	aRRS	PAR
$n = 10$	2.18	3.67	**10.71**	**30.15**	18.73	6.81	31.19	**31.33**	30.85	2.16	3.27	**6.22**	1.19	1.43	**2.21**
$n = 20$	3.74	6.16	**16.89**	**29.02**	17.95	6.57	**32.40**	32.15	**32.40**	3.63	5.17	**8.46**	1.84	2.25	**3.37**
$n = 50$	7.17	11.28	**27.76**	**27.61**	16.02	6.17	**38.45**	37.85	38.40	6.52	8.37	**11.66**	3.24	3.87	**5.53**
$n = 100$	12.72	19.64	**39.41**	**25.99**	13.72	5.66	**44.10**	43.55	43.95	10.24	12.52	**14.74**	5.54	6.51	**8.44**
$n = 200$	18.67	28.65	**50.98**	**23.77**	12.10	5.25	**47.75**	**47.75**	47.65	14.22	16.67	**17.82**	8.23	9.61	**11.51**
$n = 500$	29.31	44.10	**66.60**	**20.12**	9.77	4.81	**50.30**	49.95	50.15	19.84	21.58	**22.02**	12.39	14.02	**15.91**
$n = 1000$	40.38	58.17	**77.63**	**16.88**	8.12	4.50	**52.55**	51.80	52.35	24.22	24.52	**25.21**	16.10	17.56	**19.23**
$n = 10000$	80.50	95.19	**96.68**	**7.29**	4.25	3.85	57.45	**57.50**	38.20	**35.20**	32.81	32.26	**29.98**	28.48	26.45

Table 1. Evaluation results for the c35k collection: Both re-ranking approaches (aRRS and PAR) are compared against the text-only RRS approach for different numbers of maximum considered top-ranked pages n. For both aRRS and PAR, we set $k = 50$, for PAR, α is also set to 50. Values (given in %) are obtained by averaging over 200 evaluation queries.

	Recall			**Precision**			**Prec@10**			**r-Precision**			**Avg. Prec. (MAP)**		
Baseline	100.00			13.32			13.33			13.31			14.31		
	Web only		PAR	Web only		PAR	Web only		PAR	Web only		PAR	Web only		PAR
PseudoDoc	81.15		**98.83**	**14.50**		13.34	30.72		**31.15**	25.77		**26.28**	22.66		**25.74**
	RRS	aRRS	PAR	RRS	aRRS	PAR	RRS	aRRS	PAR	RRS	aRRS	PAR	RRS	aRRS	PAR
$n = 10$	5.96	**62.23**	59.08	**25.77**	14.49	14.84	**25.77**	23.81	23.74	5.61	18.35	**18.44**	3.58	**14.26**	13.51
$n = 20$	10.19	**80.90**	75.90	**24.87**	13.99	14.34	25.98	**26.40**	25.61	8.84	20.55	**20.70**	5.30	**18.36**	17.20
$n = 50$	17.99	**93.45**	89.52	**22.84**	13.33	13.63	26.06	**27.84**	26.04	13.49	**22.92**	22.23	7.57	**21.55**	20.19
$n = 100$	26.80	**96.60**	94.78	**21.02**	13.15	13.39	29.30	**30.07**	29.28	18.05	**23.88**	23.79	10.59	**23.07**	22.41
$n = 200$	38.63	**97.23**	96.38	**19.15**	13.08	13.22	30.60	**31.58**	30.79	21.58	24.32	**24.38**	13.84	**24.27**	23.90
$n = 500$	56.31	**97.37**	97.05	**16.86**	13.07	13.15	32.68	32.52	**33.17**	24.06	25.79	**25.91**	18.02	25.19	**25.27**
$n = 1000$	66.91	**97.47**	97.18	**15.54**	13.06	13.13	33.47	33.45	**33.60**	24.86	25.90	**26.52**	20.37	**25.85**	25.64
$n = 10000$	73.27	**97.61**	97.31	**14.56**	13.05	13.13	33.62	**33.74**	33.67	25.06	**26.95**	26.76	21.77	**26.58**	25.82

Table 2. Evaluation results for the CAL500 set: Values are obtained by averaging over 139 evaluation queries. Apart from that, the same settings as in Table 1 are applied.

not be used. Hence, we remove all negating tags. Furthermore, we join redundant tags (mostly genre descriptors). For tags consisting of multiple descriptions (e.g., *Emotion-Emotional/Passionate*) we use every description as independent query. This results in a total set of 139 test queries.

4.3 Evaluation Measures and Results

To measure the quality of the obtained rankings and the impact of the combination approaches, we calculate standard evaluation measures for retrieval systems, cf. [15]. Table 1 shows the results for the c35k collection (averaged over all 200 queries): The top row contains the baseline that has been empirically determined by repeated evaluation of random permutations of the collection. Not unexpectedly, the incorporation of additional tracks via the audio similarity measure leads to an increase in overall recall while precision is worsened. However, these global measures are not too important since for rankings one is in general more interested in how fast (i.e., at which position in the ranking) relevant results are returned. To this end, measures like *Precision @ 10 documents*, *r-Precision* (i.e., precision at the r^{th} returned document, where r is the number of tracks relevant to the query), and *(mean) average precision* (i.e., the arithmetic mean of precision values at all encountered relevant documents) give more insight into the quality of a ranking. For r-Precision and average precision we can clearly see that PAR (and also aRRS) perform better than text-based RRS. However, when comparing this to the pseudo-document indexing approach, we see that this simple and efficient ranking technique is in most cases even better than the combination with audio.[3] Thus, although audio similarity may improve results, it can not keep up to a well working text-based approach. Furthermore, we can see that incorporation of audio worsens results if recall of the initial ranking is already high (n=10000, PseudoDoc). The reason is that audio similarity introduces a lot of noise into the ranking. Hence, to preserve the good performance at the top of the rankings, α should be set to a high value. On the other hand, this prevents theoretically possible improvements. For the CAL500 set (Table 2), things look a bit different. Here, the aRRS approach performs clearly superior to RRS. Improvements can even be observed within the first ten documents. For this collection, also results of the PseudoDoc approach can be improved by applying post-hoc audio-based re-ranking. For comparison of different retrieval strategies, we calculated *precision at 11 standard recall levels*. For each query, precision $P(r_j)$ at the 11 standard recall levels $r_j, j \in \{0.0, 0.1, 0.2, ..., 1.0\}$ is interpolated according to $P(r_j) = max_{r_j \leq r \leq r_{j+1}} P(r)$. This allows averaging over all queries and results in character-

[3] Note that the *Web only* recall value of PseudoDoc represents the upper bound for all purely text-based approaches.

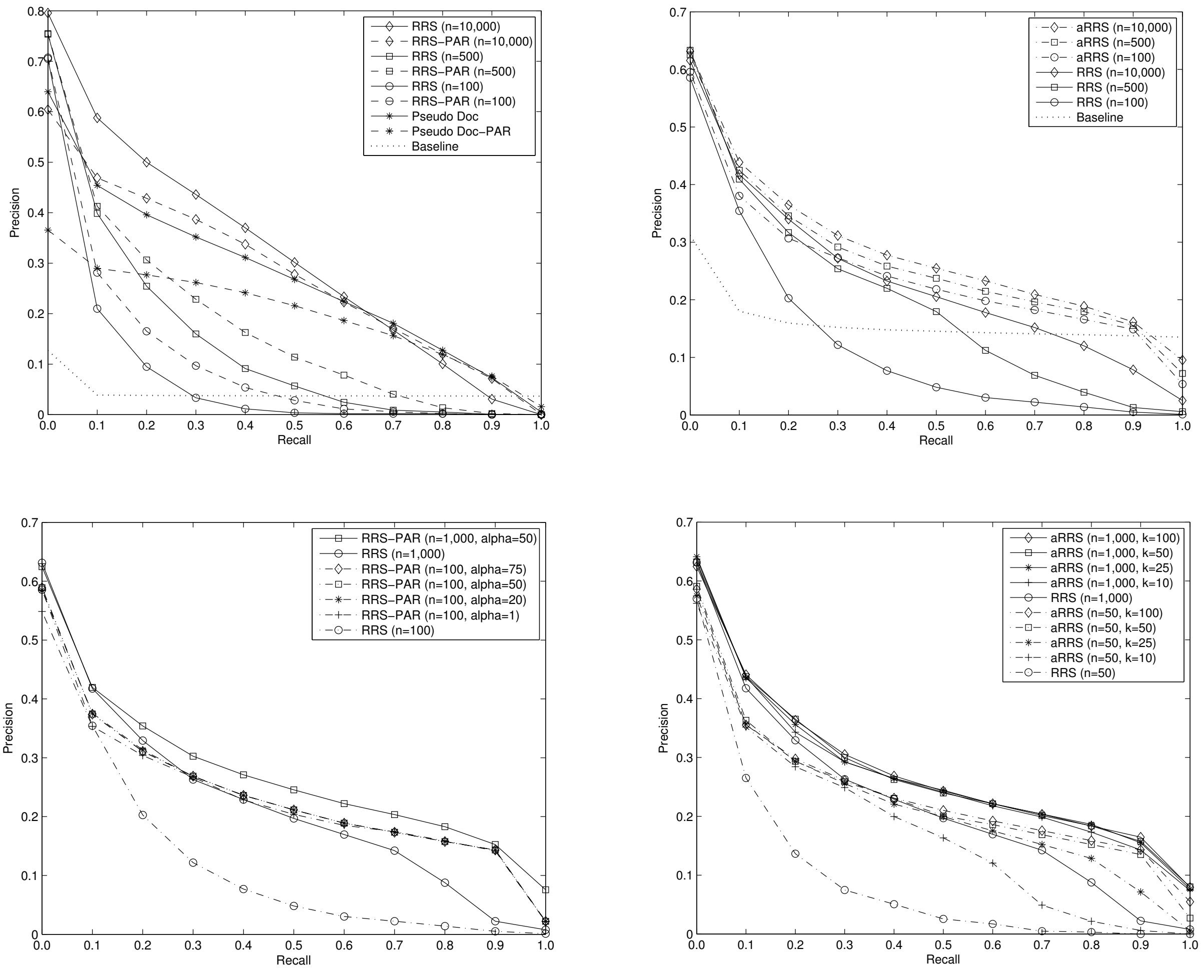

Figure 1. *Precision at 11 Standard Recall Levels* plots: The upper left plot depicts selected curves (averaged over all queries) from evaluating the c35k set for comparison of the RRS approach and subsequent PAR re-rankings. The upper right plot depicts (averaged) curves from the CAL500 set for comparison of the RRS and the aRRS approaches. The lower figures are intended to give an impression of the effects of different parameters for PAR (left) and aRRS (right). Both are calculated on the CAL500 set.

istic curves for each retrieval algorithm, enabling comparison of distinct algorithms/settings. Figure 1 depicts several precision at 11 standard recall level curves. The two plots at the top basically confirm what could be seen in tables 1 and 2. The two plots at the bottom show the influence of parameters α and k on the retrieval quality.

Using the CAL500 set, we can (rather informally) evaluate how audio similarity influences retrieval of tracks from the so-called "long tail". To this end, we restricted the set of relevant tracks for each query to contain only tracks from the (in general not so well known) online record label Magnatune. Absolute numbers resulting from this type of evaluation are rather discouraging, however, when comparing the results from RRS_{200} with those from $aRRS_{200}$ on this modified ground truth, a small improvement can be observed (e.g., MAP increases from 2.03 to 3.68, rPrec from 2.44 to 2.82). Optimistically spoken, a positive tendency is recognizable – from a more realistic perspective, both results are disappointing. In any case, the impact on long tail tracks needs a thorough investigation in future work.

5. DISCUSSION

We have shown that combining Web-based music indexing with audio similarity has the potential to improve retrieval performance. On the other side, we have also seen that even an improved combined retrieval approach may be outperformed by another, rather simple, text-only approach. Possible explanations are inadequate combination functions and/or an inadequate audio similarity measure. To estimate the potential of the audio similarity measure for this task, we examined the 100 nearest audio neighbors for every relevant track for a query and for every query, i.e., at each position $k = 1...100$, we calculated the precision (wrt. the currently examined query). Figure 2 shows the result averaged over all seed songs and queries for the c35k collection. Within the top 10 neighbors, a precision of around 7% can be expected in average based solely on the audio similarity. However, it is questionable whether this can be improved as audio similarity measures (statically) focus on specific musical properties, whereas textual

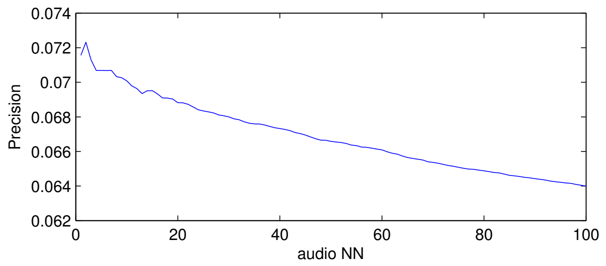

Figure 2. Precision at audio-based nearest neighbor for the c35k set (averaged over all queries; for every query average of rankings with each relevant track as seed).

queries can be aimed at basically every aspect of music, from different acoustic properties, to cultural context, to completely unrelated things.

In general it has to be stated that proper combination of these two sources is rather difficult since they target different directions and applications. Furthermore, a combination function can not be optimized in advance to suit every potential query, i.e., in contrast to, e.g., [5], automatic learning of proper combination functions (e.g., via machine learning methods) is not applicable for this task since we have no learning target. More precisely, Web-based music indexing as we currently apply it is an unsupervised approach. This is implied by the requirement to deal with a large and arbitrary vocabulary.

6. CONCLUSIONS AND FUTURE WORK

We proposed two methods to combine a Web-based music retrieval system with an audio similarity measure to improve overall ranking results and enable including tracks not present on the Internet into search results. Based on our evaluations, we could show that the overall ranking quality can be improved by integrating purely acoustic similarity information. However, we were also confronted with the current limitations of this combination. The first results gathered, open up new questions for future work, e.g., if another audio similarity measure could produce more substantial results. Also the question of combining the different sources will be taken a step further. Possible future enhancements could comprise clustering to find coherent groups of songs. This could be based on learning from many queries and finding stable relations between frequently co-occurring tracks. Another aspect that will be dealt with in future work is the impact on tracks from the long tail. Ideally, a combination would allow for retrieval of relevant tracks irrespective of their presence on the Web.

7. ACKNOWLEDGMENTS

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L511-N15. Peter Knees would like to thank the staff of ChEckiT! - Verein Wiener Sozialprojekte (where he carried out his civilian national service) for tolerating and supporting the development of this publication.

8. REFERENCES

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

[2] O. Celma, P. Cano, and P. Herrera. Search Sounds: An audio crawler focused on weblogs. *Proc. 7th ISMIR*, 2006.

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

[4] M. Sordo, C. Laurier, and O. Celma. Annotating music collections: How content-based similarity helps to propagate labels. *Proc. 8th ISMIR*, 2007.

[5] L. Barrington, D. Turnbull, M. Yazdani, and G. Lanckriet. Combining audio content and social context for semantic music discovery. *Proc. 32nd ACM SIGIR*, 2009.

[6] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, and K. Seyerlehner. A Document-centered Approach to a Natural Language Music Search Engine. *Proc. 30th ECIR*, 2008.

[7] O. Gospodnetić and E. Hatcher. *Lucene in Action*. Manning, 2005.

[8] D. Turnbull, L. Barrington, and G. Lanckriet. Five approaches to collecting tags for music. *Proc. 9th ISMIR*, 2008.

[9] T. Pohle and D. Schnitzer. Striving for an Improved Audio Similarity Measure. *4th MIREX*, 2007.

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

[11] E. Pampalk. *Computational Models of Music Similarity and their Application to Music Information Retrieval*. PhD thesis, Vienna University of Technology, 2006.

[12] A. Flexer. A closer look on artist filters for musical genre classification. *Proc. 8th ISMIR*, 2007.

[13] C. Snoek, M. Worring, and A. Smeulders. Early versus late fusion in semantic video analysis. *Proc. 13th ACM Multimedia*, 2005.

[14] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, 16(2):467–476, February 2008.

[15] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Reading, Massachusetts, 1999.