SUPERVISED AND UNSUPERVISED WEB DOCUMENT FILTERING TECHNIQUES TO IMPROVE TEXT-BASED MUSIC RETRIEVAL

Peter Knees, Markus Schedl, Tim Pohle, Klaus Seyerlehner, and Gerhard Widmer
Department of Computational Perception
Johannes Kepler University Linz, Austria
peter.knees@jku.at

ABSTRACT

We aim at improving a text-based music search engine by applying different techniques to exclude misleading information from the indexing process. The idea of the original approach is to index music pieces by "contextual" information, more precisely, by all texts to be found on Web pages retrieved via a common Web search engine. This representation allows for issuing arbitrary textual queries to retrieve relevant music pieces. The goal of this work is to improve precision of the retrieved set of music pieces by filtering out Web pages that lead to irrelevant tracks. To this end we present two unsupervised and two supervised filtering approaches. Evaluation is carried out on two collections previously used in the literature. The obtained results suggest that the proposed filtering techniques can improve results significantly but are only effective when applied to large and diverse music collections with millions of Web pages associated.

1. MOTIVATION AND CONTEXT

Searching for music by issuing "semantic" and descriptive queries has become a hot research topic recently [2, 4, 8, 11–13, 15]. While typical *query-by-example* systems require the user to have a specific piece of music at hand (or at least in mind) when searching for other music, *query-by-description* systems allow for typing in a short characterisation or a related term to find desired music. Moreover, it is generally desirable to build systems that are capable of linking music to meaningful textual descriptions (i.e., bridging what is often misleadingly called "semantic gap" [17]). For instance, this capability can be used to recommend music based on other textually represented information, e.g., by analysing the user's currently viewed Web page [7].

For the dedicated task of building a music search engine, several approaches have been presented. In [4], Baumann et al. describe a system incorporating various kinds of meta-data, lyrics, and acoustic properties. To analyse queries, natural language processing methods and knowledge from a semantic ontology are applied to map the query tokens to the corresponding concepts. In [8], Celma et al. propose usage of a Web crawler focused on audio blogs to obtain textual descriptions for music. Blog entries are extracted and the associated music pieces are indexed based on this information. From a text-based retrieval result, also acoustically similar songs can be discovered. Yang et al. [18] index a music collection using lyrics and apply a combination of text and audio descriptors to cluster results.

In [15,16], Turnbull et al. present a method for semantic retrieval. Based on the CAL500 data set – a collection of 500 songs manually labelled with descriptions representing music-relevant properties – models of these properties are learned from audio features. The system can then be used to retrieve relevant songs based on queries consisting of the words used for annotation. In [2], this approach is extended by incorporating multiple sources of features (i.e., acoustic features related to timbre and harmony, social tags, and Web documents). These largely complementary sources are combined to improve prediction accuracy.

In [13], we propose an unsupervised strategy for music retrieval that is capable of dealing with a large and arbitrary vocabulary. Contrary to learning a pre-defined set of labels (cf. [2, 15]), music pieces are represented in a vector space constructed from related Web documents. An improved version of this approach is presented in [11]. Instead of aggregating Web pages to construct term vectors, the retrieved Web documents are stored in an index. A given query is processed by passing the query to this index and applying a technique called *rank-based relevance scoring* to the resulting document ranking. This scoring is based on the associations between music tracks and Web documents (as we further extend this approach in this paper, a more detailed description can be found in Section 2). In [12], we propose unsupervised methods to improve search results by integrating audio similarity. Results show that the combinations can raise performance slightly but mainly introduce noise.

With this work, we aim at enhancing the approach from [11] by constructing filters that remove misleading information from the Web document index and raise precision of the retrieved music piece rankings (cf. [3]). Two of these filters are built in an unsupervised manner, whereas the other two make use of external annotations for learning to distinguish between informative and noisy content.

2. WEB-BASED MUSIC INDEXING

In the approach from [11, 12], music is indexed by using "contextual" meta-information about the pieces under consideration. This context-data is assumed to be found on related Web pages retrieved via Google by issuing three queries for each music piece m. Constructed from the meta-information categories *artist name*, *album name*, and *track title*, these queries are *"artist" music*, *"artist" "album" music review -lyrics*, and *"artist" "title" music review -lyrics*. For each of these queries, the top-100 Web pages are retrieved and joined into D_m, the set of pages associated with m. All retrieved documents are also stored in an index I. Relevance of a music piece m wrt. a given query q is assessed by querying I with q and applying *rank-based relevance scoring (RRS)* to the n most relevant Web documents in I (Equation 1).

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

In this equation, n denotes the maximum number of top-ranked documents when querying I, $D_{q,n}$ the ordered set (i.e., the ranking) of the n most relevant Web documents in I with respect to query q, and $rnk(p, D_{q,n})$ the rank of document p in $D_{q,n}$. For retrieval, the final ranking of music tracks is obtained by sorting the music pieces according to their RRS value.

In the published evaluations [11, 12], precision hardly ever exceeds 30% using this scoring approach, i.e., rankings usually contain three times more irrelevant music pieces than relevant. Based on this, also subsequent steps such as combination with audio similarity may suffer from erroneous input. Clearly, the reason for the high number of irrelevant pieces has to be searched for in the underlying Web pages. For indexing, all pages returned by Google are considered relevant, irrespective of whether they actually contain information about or descriptions of the corresponding music piece or artist. Furthermore, the page indexer does not distinguish between text that occurs in the "main part" of the Web page and text that is used for navigation or links to stories about other, completely unrelated artists. Thus, to improve precision of the retrieved set of music pieces, in the next section, we propose four different filtering approaches to remove noisy information and documents.

3. DOCUMENT FILTERING TECHNIQUES

This section describes the proposed filtering methods to exclude noisy information from the indexing process. We explore two types: Unsupervised and supervised filters.

3.1 Unsupervised Filtering

The characteristic of these filters is that they aim at identifying misleading texts without information from external sources. Hence, they can be applied to the index directly after building it. The first filter does not remove full documents from the index, but tries to identify those portions within the indexed text that do not contain specific information. The second approach identifies and removes complete documents.

3.1.1 Alignment-Based Noise Removal

As mentioned earlier, most indexed Web pages do not only contain relevant and interesting information (if any at all). Almost every page contains a site-specific header, navigation bar, links to related pages, and copyright disclaimers, frequently automatically generated by a content management system, cf. [9, 19]. Especially on music pages, these segments often feature lists of other music artists, genres, or tag clouds to facilitate browsing. This surrounding information is usually not relevant to the associated music piece and should thus be ignored.

Removal of this kind of text is the aim of this filter. Since large parts of the surrounding text remain the same for most pages within a Web domain, we can identify redundant segments by comparing several texts from the same domain. Coherent parts are most likely to be non-specific for a given music piece and can therefore be removed. To this end, we adopt the *multiple lyrics alignment* technique originally used to extract lyrics from multiple Web sources by matching coherent parts and preserving overlapping segments [14]. In the current filtering scenario, the overlapping segments are going to be removed.

To apply the filter, we collect all documents belonging to the same domain. Since for many blogs, the domain alone does not indicate similarly structured pages – different blogs are typically accessible via separate subdomains (e.g., for blogspot.com) – we keep the subdomain if the host section of the URL contains the word "blog". For domains that occur only up to five times in the page index, no filtering is performed. For all other domains up to eight documents are chosen randomly and used for alignment. From the alignment, we choose all aligned tokens occurring in at least 60% of the aligned texts and finally select all text sequences consisting of at least 2 tokens. The resulting sequences are then removed in all Web pages originating from the domain.

3.1.2 Too-Many-Artists Filtering

With this filter, the goal is to detect pages that do not deal with only one type of music, i.e., pages that provide an ambiguous content and are therefore a potential source of error. Some of these pages can be identified easily, since they contain references to many artists. Hence, we query the page index with all artist names from the music collection and count the occurrences of each page in the result sets. Constructing the filter simply consists in selecting a threshold for the maximum number of allowed artists per page. By systematically experimenting with this threshold, we yielded most promising results when removing all pages containing more than 15 distinct artists. Throughout the rest of this paper, *too-many-artists filtering* refers to the removal of pages containing more than 15 artists.

3.2 Supervised Filtering

As already mentioned in [12], automatic optimisation of the (unsupervised) Web-based indexing approach is difficult, since for arbitrary queries, there is no learning target known in advance (in contrast, for instance, to the approaches presented in [2, 15], where the set of possible queries is limited). However, in terms of identifying sources of noise, automatic optimisation approaches are somewhat more promising, provided that a set of potential queries with corresponding relevance judgements is available. The idea is that by observing performance on a given set of queries, it should be possible to learn to identify and exclude misleading Web pages and therefore yield better results also on other, previously unseen queries. This is based on the assumption that documents responsible for introducing noise to a music piece ranking contain erroneous (at least ambiguous) information and are likely to introduce noise to other queries too.

3.2.1 Query-Based Page Blacklisting

Following the general idea outlined in Section 3.2, we construct a simple filter that blacklists (i.e., excludes) Web pages contributing more negatively than positively to query results. Hence, based on RRS we calculate a simple score to rate a page p:

$$S_n(p) = \sum_{q \in Q} \left(\sum_{m \in M_p \cap T_q} RRS_n(m,q) - \sum_{m \in M_p \cap \overline{T_q}} RRS_n(m,q) \right) \quad (2)$$

where Q denotes the set of all available queries/annotations, M_p the set of all music pieces associated with page p, T_q the set of all pieces annotated with q (i.e., relevant to query q), and $\overline{T_q}$ its complement (i.e., all music pieces not relevant to q). Informally speaking, over all queries, we subtract the sum of RRS scores contributed to negative examples from the sum of RRS scores contributed to positive examples. We then remove all Web documents p with $S_n(p) < 0$, i.e., all documents that contributed more negatively than positively over the course of all queries.

3.2.2 Query-Trained Page Classification

While the *query-based page blacklisting* filter represents (if any) just the "laziest" form of machine learning (i.e., merely recognising instances without any kind of generalisation), this filter aims at learning to automatically classify Web pages as either "positive" (keep) or "negative" (remove). Hence, it should be better suited to deal with new queries that provoke previously unseen (and thus unrated) Web pages. To get positive and negative examples as training instances for the classifier, only pages that have either contributed exclusively positively or exclusively negatively are considered. Positive examples are defined as $\{p \,|\, p \in D_{q,n}, \forall q \in Q : M_p \cap \overline{T_q} = \emptyset\}$ and negative as $\{p \,|\, p \in D_{q,n}, \forall q \in Q : M_p \cap T_q = \emptyset\}$ (cf. Eq. 2). As a further requirement, only pages that appear in at least two query result sets are considered. As feature representation for Web pages, we incorporate characteristic values such as the length of the page's (unparsed) HTML content, the length of the parsed content, the number of different terms occurring on the page, the number of associated music pieces (i.e., $|M_p|$), the number of contained artist names (cf. 3.1.2), as well as ratios between these numbers. Furthermore, we utilise title and URL of the pages as very short textual representations that are converted into a term vector space (using the functions provided by WEKA [10]) and added as numerical features.

For classification, we decided to use the *Random Forest Classifier* [5] from the WEKA package (with 10 trees). Since there are usually significantly more negative than positive examples, we also apply a cost-sensitive meta-classifier to raise importance of positive instances (misclassification of positive instances is penalised by the ratio of negative to positive examples).

4. EVALUATION

For evaluation of the different filtering approaches, we use both test collections from [12]. The first collection, called c35k, is a large real-world collection and contains 35,000 mostly popular pieces. For evaluation purposes, a benchmarking set consisting of 200 queries and relevance judgements has been created from Last.fm tags [1] . The second collection is the CAL500 set, a collection of 500 songs manually labelled with words representing various music-relevant properties [15]. For comparison, we adopted the 139 category subset used in [12]. To test effectiveness of the retrieval approaches, annotations are used as queries to the system. They also serve as relevance indicator, i.e., a track is considered to be relevant for query q if it has been tagged with tag q. For evaluation of the supervised filtering approaches, a 10-fold cross validation is performed on the test collections, i.e., in each fold, 90% of the queries are used to train the filters which are then applied and evaluated on the remaining 10%.

To measure the quality of the obtained rankings, standard evaluation measures for retrieval systems are calculated, cf. [1]. Additionally to the "global" measures *precision* and *recall*, ranking 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* (*MAP*, i.e., the arithmetic mean of precision values at all encountered relevant documents) are used for evaluation. To further compare different retrieval strategies, we calculate *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 characteristic curves for each retrieval algorithm, enabling comparison of distinct settings. To obtain a single value for comparison of these curves, we calculate the area under the curve (*prec@11std.recall - AUC*). For presentation of the c35k collection, we decided to use tables instead of graphs to show more detailed results (including significance tests).

[1] `http://www.last.fm`

	Recall						**Precision**					
	UNF	ANR	2MA	A+2	QPB	QPC	UNF	ANR	2MA	A+2	QPB	QPC
$n = 10$	**2.18**	**2.01**	**2.07**	1.95	**2.01**	**2.69**	30.15	**31.89**	**35.72**	30.71	**34.32**	**36.98**
$n = 20$	**3.74**	**3.95**	**3.93**	**3.66**	**3.89**	**4.93**	29.02	31.30	**32.86**	30.91	**33.91**	**37.11**
$n = 50$	**7.17**	7.48	**7.87**	7.34	**8.15**	**9.15**	27.61	29.50	**31.13**	28.56	**32.50**	**34.79**
$n = 100$	**12.72**	12.09	12.58	**11.92**	**12.80**	**14.26**	25.99	27.64	29.05	27.25	**31.75**	**32.42**
$n = 200$	**18.67**	18.22	18.13	**17.95**	19.08	**20.73**	23.77	25.77	**26.21**	25.26	**28.60**	**28.74**
$n = 500$	29.31	29.60	29.84	28.16	29.96	**32.00**	20.12	21.69	21.71	21.00	**24.76**	**23.19**
$n = 1,000$	40.38	40.31	40.11	36.41	40.43	**41.52**	16.88	17.86	18.19	18.00	**20.75**	18.92
$n = 10,000$	**80.50**	**79.56**	76.80	57.55	73.50	73.32	7.29	7.42	7.87	**11.90**	**9.63**	8.62

	Prec@10						**r-Precision**					
	UNF	ANR	2MA	A+2	QPB	QPC	UNF	ANR	2MA	A+2	QPB	QPC
$n = 10$	**31.19**	**34.94**	**37.76**	**32.74**	**36.45**	**37.32**	**2.16**	**1.99**	**2.05**	1.79	**2.01**	**2.68**
$n = 20$	**32.40**	**34.96**	**37.05**	**31.93**	**36.75**	**37.89**	**3.63**	**3.68**	**3.67**	**3.39**	**3.69**	**4.57**
$n = 50$	**38.45**	36.20	**41.70**	36.52	**40.25**	**43.90**	**6.52**	6.85	**7.08**	6.29	**7.30**	**8.38**
$n = 100$	**44.10**	39.05	**46.65**	40.52	**43.40**	**48.40**	**10.24**	10.41	10.78	10.27	**11.52**	**12.48**
$n = 200$	47.75	42.15	**48.90**	43.82	**46.95**	**49.60**	14.22	14.54	14.68	14.27	**16.03**	**16.83**
$n = 500$	**50.30**	45.95	**51.20**	47.32	**49.75**	**52.45**	19.84	21.01	20.35	19.85	**22.54**	**23.52**
$n = 1,000$	**52.55**	48.75	**52.85**	48.47	**53.15**	**54.20**	24.22	25.43	25.00	23.66	**27.48**	**27.85**
$n = 10,000$	**57.45**	57.20	**56.70**	50.12	**62.35**	**58.00**	**35.20**	**35.77**	**35.03**	28.28	**35.69**	**34.70**

	Avg. Prec (MAP)						**Prec@11Std.Recall - AUC**					
	UNF	ANR	2MA	A+2	QPB	QPC	UNF	ANR	2MA	A+2	QPB	QPC
$n = 10$	**1.19**	**1.32**	**1.38**	**1.12**	**1.39**	**1.83**	**3.05**	**3.15**	**3.34**	**2.95**	**3.23**	**3.61**
$n = 20$	1.84	2.14	**2.24**	1.86	**2.26**	**2.95**	**3.64**	3.98	**4.07**	**3.74**	**4.06**	**4.78**
$n = 50$	3.24	3.79	**4.06**	3.58	**4.49**	**5.22**	4.99	5.63	**5.86**	**5.44**	**6.39**	**6.67**
$n = 100$	5.54	5.93	**6.29**	5.67	**7.00**	**7.64**	7.11	7.56	**7.91**	7.34	**8.51**	**9.14**
$n = 200$	8.23	8.61	8.78	8.26	**10.24**	**10.65**	9.62	10.12	10.19	9.75	**11.72**	**12.20**
$n = 500$	12.39	13.30	13.19	12.38	**15.06**	**15.93**	13.76	14.69	14.57	13.89	**16.45**	**17.43**
$n = 1,000$	16.10	17.37	17.01	15.45	**19.41**	**19.84**	17.22	18.84	18.36	16.82	**20.82**	**21.03**
$n = 10,000$	**29.98**	**30.60**	**29.80**	21.86	**30.92**	29.22	**31.25**	**31.84**	**31.07**	23.07	**32.05**	30.39

Table 1. Comparison of *unfiltered RRS* (UNF) vs. *Alignment-Based Noise Removal* (ANR), *Too-Many-Artists Filtering* (2MA), *ANR+2MA* (A+2), *Query-Based Page Blacklisting* (QPB), and *Query-Trained Page Classification* ($\text{QPC}_{n=200}$) for the c35k collection and different values of n, i.e., the maximum number of retrieved Websites incorporated in RRS. QPB and QPC are performed upon ANR. Values (given in %) are obtained by averaging over 200 evaluation queries (for supervised approaches via 10-fold Cross Validation). Entries in bold face indicate that there is no significant difference between this entry and the best performing, i.e. bold entries indicate the "best group" (Friedman test, $\alpha = 0.01$). Note that due to the rank-based nature of the non-parametric Friedman test, results may belong to the best group even with lower average values than significantly worse results.

Table 1 shows evaluation results on the c35k collection for different values of n (number of top ranked Web documents when querying the page index). For the alignment-based noise removal (ANR), we observe slight improvements for the averaged results especially for precision, r-precision, average precision and the area under the standardized precision-recall curve. However, in the Friedman test these results are not significant. For recall and precision@10 we can see a significant drop in performance.

The too-many-artists filter (2MA) outperforms the unfiltered RRS significantly in terms of precision and average precision for smaller values of n. A decrease is most clearly visible for recall. In addition, we evaluated also the combination of both unsupervised filters (A+2). In most cases, this combination worsens results significantly which is rather surprising, considering that these filters target different levels of noise removal. However, it seems that too much information is excluded when using both.

Except for recall, both *supervised approaches* are constantly in the best performing group, superiority is clearly visible for precision. For the query-trained page classification filter (QPC), it has to be mentioned that for values of $n > 500$ the number of training instances gets very high, slowing down the evaluation progress. For this reason, we decided to use the QPC filter with the $n = 200$ setting also for experiments with $n \neq 200$. This explains also the slight drop for $n \geq 1,000$. Still, results are more than acceptable for QPC.

	Recall						Precision					
	UNF	ANR	2MA	A+2	QPB	QPC	UNF	ANR	2MA	A+2	QPB	QPC
$n = 10$	5.96	6.10	5.85	6.00	6.10	6.69	25.77	27.17	25.72	26.99	27.22	28.35
$n = 20$	10.19	9.68	9.46	9.49	9.83	10.69	24.87	25.39	24.69	25.44	25.07	25.73
$n = 50$	17.99	18.09	17.20	17.61	18.23	17.89	22.84	23.14	22.44	22.94	23.22	22.68
$n = 100$	26.80	26.34	25.48	25.90	27.11	***24.34***	21.02	21.05	20.80	21.10	21.69	20.16
$n = 200$	38.63	38.62	37.24	37.21	37.95	***31.75***	19.15	19.30	19.11	19.25	19.67	18.54
$n = 500$	56.31	55.65	54.26	***53.31***	***51.21***	***38.09***	16.86	16.95	16.92	17.05	17.74	17.24
$n = 1,000$	66.91	66.48	63.78	***62.69***	***53.10***	***40.10***	***15.54***	***15.81***	***15.74***	***16.01***	17.36	16.74
$n = 10,000$	73.27	72.93	***69.06***	***68.02***	***37.98***	***40.76***	***14.56***	***14.84***	***14.82***	***15.17***	20.75	***16.56***

Table 2. Comparison of unfiltered RRS vs. the filter approaches for the CAL500 set averaged over 139 queries (cf. Table 1). Note that in contrast to Table 1, in this table, bold and italic appearing entries indicate a significant difference to the group of best approaches, i.e., worse results are marked. For all experiments with QPC, the setting $\text{QPC}_{n=50}$ is used.

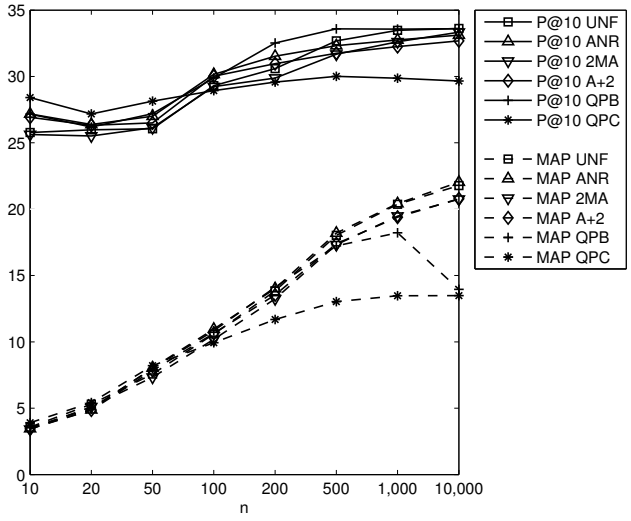

Figure 1. *Precision@10* (upper curves) and *Avg. Prec (MAP)* (lower curves) for the CAL500 set and different values of n (cf. Table 2).

For the CAL500 set, results are very disappointing (Table 2). No proposed filter can significantly improve results (except for precision of the supervised filters with high values of n, which go along with a dramatic loss in recall due to a very high number of excluded pages). The reasons are not directly comprehensible. One possibility could be that in the case of the c35k set with associated Last.fm tags, the approaches benefit from the inherent redundancies in the tags/queries (e.g., *metal* vs. *black metal* vs. *death metal*). In the case of the CAL500 set, queries exhibit no redundancy, as the set is constructed to describe different dimensions of music. However, this would only affect the supervised filters.

Another explanation could be that the CAL500 page index contains considerably less pages than the c35k index (approx. 80,000 vs. approx. 2 million pages). First, and also in the light that the CAL500 set has been carefully designed, it seems possible that the index does not contain so much noise. Hence, the proposed noise removal strategies don't work here. Second, since the index is rather small, removal of a relatively high number of pages has a higher impact on the overall performance. This becomes especially apparent when examining the results of the supervised approaches for high n. Apart from the results, it should be noted that the CAL500 set is without doubt very valuable for research (high quality annotations, freely available, etc.) but at the same time, it is a highly artificial corpus which can not be considered a "real-world" collection. Hence, some "real-world" problems maybe can not be tested with such a small set.

5. CONCLUSIONS AND FUTURE WORK

We have demonstrated the usefulness of two unsupervised and two supervised filtering approaches for Web-based indexing of music collections. Evaluation showed inconsistent results for two collections with very different characteristics and suggests that the proposed filtering techniques can improve results significantly when applied to large and diverse music collections with millions of Web pages associated.

Regarding the proposed filtering techniques, more or less all of them proved to be useful and could improve not only the overall precision but also the ranking of music pieces. By introducing supervised optimisation into this originally unsupervised technique, there is still more potential to tweak performance. For instance, we are convinced that a more carefully selected feature set can easily improve results of page classification further. Using annotated sets for learning, also proper combination with audio similarity, e.g., to raise recall, could be possible.

Instead of finding redundant portions in Web pages from the same domain by aligning and matching their content, techniques like *vision page segmentation* [6] could help in identifying the relevant parts of a Web page. By extracting smaller segments from Web pages, the principle of the RRS weighting could be transferred to "blocks" and scoring could be designed more specifically.

Another aspect not directly related to filtering pages became apparent during experiments with the Too-many-artists filter. When querying the page index with the names

of the contained artists, artists with common speech names can be easily identified. As each artist only has a limited number of associated pages in the index, true occurrences are somehow normalised. For artists that occur much more often than expected (outlier), it can be assumed that they have common speech names. This finding could be interesting for related tasks in future work.

6. ACKNOWLEDGMENTS

This research is supported by the Austrian "Fonds zur Förderung der Wissenschaftlichen Forschung" (FWF) under project number L511-N15.

7. REFERENCES

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Reading, Massachusetts, 1999.

[2] Luke Barrington, Douglas Turnbull, Mehrdad Yazdani, and Gert Lanckriet. Combining audio content and social context for semantic music discovery. In *Proceedings of the 32nd ACM SIGIR*, Boston, MA, USA, 2009.

[3] Stephan Baumann and Oliver Hummel. Using Cultural Metadata for Artist Recommendation. In *Proceedings of the 3rd International Conference on Web Delivering of Music (WEDELMUSIC 2003)*, September 2003.

[4] Stephan Baumann, Andreas Klüter, and Marie Norlien. Using natural language input and audio analysis for a human-oriented MIR system. In *Proceedings of the 2nd International Conference on Web Delivering of Music (WEDELMUSIC 2002)*, Darmstadt, Germany, 2002.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting content structure for web pages based on visual representation. In *Fifth Asia Pacific Web Conference (APWeb 2003)*, 2003.

[7] Rui Cai, Chao Zhang, Chong Wang, Lei Zhang, and Wei-Ying Ma. Musicsense: contextual music recommendation using emotional allocation modeling. In *Proceedings of the 15th ACM Multimedia*, 2007.

[8] Oscar Celma, Pedro Cano, and Perfecto Herrera. Search Sounds: An audio crawler focused on weblogs. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, Victoria, B.C., Canada, 2006.

[9] Sandip Debnath, Prasenjit Mitra, and C. Lee Giles. Automatic extraction of informative blocks from webpages. In *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC'05)*, 2005.

[10] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[11] Peter Knees, Tim Pohle, Markus Schedl, Dominik Schnitzer, and Klaus Seyerlehner. A Document-centered Approach to a Natural Language Music Search Engine. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR'08)*, Glasgow, Scotland, UK, March 30-April 3 2008.

[12] Peter Knees, Tim Pohle, Markus Schedl, Dominik Schnitzer, Klaus Seyerlehner, and Gerhard Widmer. Augmenting Text-Based Music Retrieval with Audio Similarity. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, Kobe, Japan, October 2009.

[13] Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer. A Music Search Engine Built upon Audio-based and Web-based Similarity Measures. In *Proceedings of the 30th ACM SIGIR*, Amsterdam, the Netherlands, July 23-27 2007.

[14] Peter Knees, Markus Schedl, and Gerhard Widmer. Multiple Lyrics Alignment: Automatic Retrieval of Song Lyrics. In *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, September 2005.

[15] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Towards Musical Query-by-Semantic-Description using the CAL500 Data Set. In *Proceedings of the 30th ACM SIGIR*, Amsterdam, the Netherlands, July 23-27 2007.

[16] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, February 2008.

[17] Geraint Wiggins. Semantic Gap?? Schemantic Schmap!! Methodological Considerations in the Scientific Study of Music. In *Proceedings of the 11th IEEE International Symposium on Multimedia (ISM'09): Workshop on Advances in Music Information Research (AdMIRe)*, 2009.

[18] Yi-Hsuan Yang, Yu-Ching Lin, and Homer Chen. Clustering for music search results. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2009.

[19] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *Proceedings of the 9th ACM SIGKDD Conference*, 2003.