INVESTIGATING THE SIMILARITY SPACE OF MUSIC ARTISTS ON THE MICRO-BLOGOSPHERE

Markus Schedl, Peter Knees, Sebastian Böck
Department of Computational Perception
Johannes Kepler University Linz, Austria
markus.schedl@jku.at, peter.knees@jku.at, sebastian.boeck@jku.at

ABSTRACT

Microblogging services such as Twitter have become an important means to share information. In this paper, we thoroughly analyze their potential for a key challenge in the field of MIR, namely the elaboration of perceptually meaningful *similarity measures*. To this end, comprehensive evaluation experiments were conducted using Twitter posts gathered during a period of several months. We investigated 23,100 combinations of different *term weighting strategies*, *normalization methods*, *index term sets*, Twitter *query schemes*, and *similarity measurement techniques*, aiming at determining in which way they influence the similarity estimates' quality.

Evaluation was performed on the task of similar artist retrieval. Two data sets were used: one of 224 well-known artists with a uniform genre distribution, the other constituting a collection of 3,000 artists extracted from last.fm and allmusic.com.

1. MOTIVATION AND CONTEXT

Term weighting techniques such as $TF \cdot IDF$ and *BM25* have been used intensely for various text retrieval tasks. Although a wealth of approaches to model the term vector space [21] on the Web has been proposed throughout the last years, e.g., [6, 12, 20, 30], IR-related research interest in the relatively novel field of microblog mining has been rather limited so far.

Microblogging has encountered a remarkable gain in popularity during the past couple of years. Being the most popular microblogging service, Twitter has more than 100 million registered users [31]. Millions of Twitter users post "tweets" that reveal what they are doing, what is on their mind, or what is currently important for them. According to [7], the number of tweets per day surpassed 50 millions in early 2010. Twitter thus represents a rich data source for text-based IE and IR.

The work at hand was inspired by [32], where the authors thoroughly evaluate various choices related to constructing text feature vectors for IR purposes, e.g., term frequency (TF), term weights (IDF), and normalization approaches. They analyze the influence of these decisions on retrieval behavior. Similarly, we present a systematic large-scale study on the influence of a multitude of decisions on music artist similarity estimation, using real-world data collections. To this end, we analyze several thousand combinations of the following single aspects: term frequency, inverse document frequency, normalization with respect to length, similarity function, index term set, and query scheme.

Elaborating musical similarity measures that are capable of capturing aspects relating to perceived similarity is one of the main challenges in MIR. Such measures enable various music applications, for example, automatic playlist generators [1], music recommender systems [4], music information systems [23], semantic music search engines [11], and intelligent user interfaces [17] to music collections.

Similarity measures based on term profiles extracted from artists' Web pages have been studied in MIR for a long time, e.g., [3, 10, 30]. In contrast, microblogs have not been harvested to a large extent so far for this purpose. To the best of our knowledge, the only work considering microblogs for similarity measurement of music artists is [24]. The authors of the aforementioned publications, however, usually select one (or a few) variant(s) of the $TF \cdot IDF$ term weighting measure and apply it to documents retrieved for music artists. The individual choices involved in selecting a specific $TF \cdot IDF$ variant and similarity function, however, do not seem to be the result of detailed assessments. In the work at hand, by contrast, we present a thorough investigation of several dimensions for modeling the music-related term vector space on the micro-blogosphere.

2. MODELING THE MICROBLOG TERM VECTOR SPACE

Similarly to the large scale experiments presented in [32], we aim at analyzing if specific combinations of the investigated algorithmic choices perform considerably better or worse than others, where performance is measured in a similarity classification task among term vector representations of tweets, cf. Section 3.
Table 1 contains an overview of the denominations used in

$\mathcal{D}$	set of documents
N	number of documents
$f_{d,t}$	number of occurrences of term t in document d
f_t	number of documents containing term t
F_t	total number of occurrences of t in the collection
$\mathcal{T}_d$	set of distinct terms in document d
f_d^m	largest $f_{d,t}$ of all terms t in d
f^m	largest f_t in the collection
$r_{d,t}$	term frequency (cf. Table 3)
w_t	inverse document frequency (cf. Table 4)
W_d	document length of d

Table 1. Denominations used in term weighting functions and similarity measures.

the different term weighting formulations (Tables 3 and 4) and similarity measures (Table 5).

2.1 Query Scheme

We decided to assess two schemes to query `Twitter` as previous work on Web-MIR [26, 30] has shown that adding music-related key terms to a search request generally improves the quality of feature vectors in terms of similarity-based classification accuracy. In Web-MIR, common terms used as additional key words are "music review" or "music genre style". Taking into account the 140-character-limitation of tweets, we decided to include only "music" as additional query term (`QS_M`) or query without any additional key terms, i.e., use only the artist name (`QS_A`) as exact phrase.

2.2 Index Term Set

Earlier work in text-based music artist modeling [9, 16, 29] shows that a crucial choice in defining the representation of an artist is that of the used index terms. For the work at hand, we hence investigated various term sets, which are summarized in Table 2. Set `TS_A` contains all terms found in the corpus (after casefolding, stopping, and stemming). Set `TS_S` is the entire term dictionary of `SCOWL` [28], which is an aggregation of several spell checker dictionaries for various English languages and dialects. Set `TS_N` encompasses all artist names present in the data set. Previous work has shown that the corresponding *co-occurrence* approach to music artist similarity estimation yields remarkable results, cf. [26]. Term set `TS_D` is a manually created dictionary of music-related terms that resembles the one used in [16]. It contains, for example, descriptors of genre, instruments, geographic locations, epochs, moods, and musicological terms. Set `TS_L` represents the most popular tags utilized by users of `last.fm`. Set `TS_F` comprises the aggregated data set for the data types *musical genre*, *musical instrument*, and *emotion*, extracted from `Freebase` [8].

To build the inverted word-level index [33], we use a modified version of the open source indexer `Lucene` [14], which we extended to represent `Twitter` posts. The extensions will be made available through our `CoMIRVA` framework [5, 25]. When creating the indexes for the different term sets, we commonly employ casefolding and stopping, e.g. [2]. Stemming, in contrast, is only performed for the term sets for which it seems reasonable, i.e., for term sets `TS_A` and `TS_S`.

2.3 TF and IDF: Term Weighting

Even though our experimental setting is guided by Zobel and Moffat's [32], we decided to extend the $TF \cdot IDF$ formulations investigated by them with *BM25*-like formulations. *BM25* is an alternative term weighting scheme, used in the *Okapi* framework for text-based probabilistic retrieval [19]. The *BM25* model includes a priori class knowledge. Since incorporating genre information into the term weighting function would bias the results of the genre classification experiments, we included an adapted formulation in the experiments, cf. variants `TF_G` and `IDF_J` in Tables 3 and 4, respectively.

2.4 Virtual Documents and Normalization

When creating a term profile from Web pages retrieved for a named entity (a music artist in our case), it is common to aggregate the pages associated with a particular entity to form a "virtual document", e.g. [3, 10]. This procedure not only facilitates handling small or empty pages, it is also more intuitive since the item of interest is the entity under consideration, not a Web page. Latest work [27] further shows that calculating term weights on the level of individual Web pages before aggregating the resulting feature vector performs inferior for the task of similarity calculation than using "virtual documents". It therefore seems reasonable to aggregate all posts retrieved from `Twitter` for an artist to one "virtual post", in particular, taking into consideration the already strong limitation of `Twitter` posts to 140 characters.

Since the different length of two artist's virtual documents is likely to influence the performance of retrieval tasks, we evaluated several normalization methods. In addition to applying no normalization (`NORM_NO`), we analyzed sum-to-1 normalization (`NORM_SUM`) and normalizing to the range $[0, 1]$ (`NORM_MAX`).

2.5 Similarity Function

The similarity measures analyzed are shown in Table 5. We included all measures investigated by Zobel and Moffat [32] that can be applied to our somewhat differing usage scenario of computing similarities between two equally dimensional term feature vectors that represent two comparable entities. We further included Euclidean similarity (`SIM_EUC`) and Jeffrey divergence-based similarity [13] (`SIM_JEF`) in the set of evaluated similarity functions.

2.6 Notation

To facilitate referring to a particular evaluation experiment, which is defined as a combination of the choices described above, we adopt the following scheme:

```
<Query Scheme>.<Index Term Set>.<Normalization>.
        <TF>.<IDF>.<Similarity Measure>
```

Abbr. / Term Set	Cardinality	Description
TS_A - all_terms	up to 1,489,459	All terms (stemmed) that occur in the corpus of the retrieved Twitter posts.
TS_S - scowl_dict	698,812	All terms that occur in the entire SCOWL dictionary.
TS_N - artist_names	224 / 3,000	Names of the artists for which data was retrieved.
TS_D - dictionary	1,398	Manually created dictionary of musically relevant terms.
TS_L - last.fm_toptags	250	Overall top-ranked tags returned by last.fm's *Tags.getTopTags* function.
TS_F - freebase	3,628	Music-related terms extracted from Freebase (genres, instruments, emotions).

Table 2. Different term sets used to index the Twitter posts.

Abbr.	Description	Formulation
TF_A	Formulation used for binary match SB = b	$r_{d,t} = \begin{cases} 1 & \text{if } t \in \mathcal{T}_d \\ 0 & \text{otherwise} \end{cases}$
TF_B	Standard formulation SB = t	$r_{d,t} = f_{d,t}$
TF_C	Logarithmic formulation	$r_{d,t} = 1 + \log_e f_{d,t}$
TF_C2	Alternative logarithmic formulation suited for $f_{d,t} < 1$	$r_{d,t} = \log_e(1 + f_{d,t})$
TF_C3	Alternative logarithmic formulation as used in *ltc* variant	$r_{d,t} = 1 + \log_2 f_{d,t}$
TF_D	Normalized formulation	$r_{d,t} = \frac{f_{d,t}}{f_d^m}$
TF_E	Alternative normalized formulation. Similar to [32] we use $K = 0.5$. SB = n	$r_{d,t} = K + (1 - K) \cdot \frac{f_{d,t}}{f_d^m}$
TF_F	Okapi formulation, according to [32]. For W we use the vector space formulation, i.e., the Euclidean length.	$r_{d,t} = \frac{f_{d,t}}{f_{d,t} + W_d / av_{d \in D}(W_d)}$
TF_G	Okapi BM25 formulation, according to [19].	$r_{d,t} = \frac{(k_1+1) \cdot f_{d,t}}{f_{d,t} + k_1 \cdot \left[(1-b) + b \cdot \frac{W_d}{av_{d \in D}(W_d)}\right]}$ $k_1 = 1.2, b = 0.75$

Table 3. Evaluated variants to calculate the term frequency $r_{d,t}$.

Abbr.	Description	Formulation
IDF_A	Formulation used for binary match SB = x	$w_t = 1$
IDF_B	Logarithmic formulation SB = f	$w_t = \log_e \left(1 + \frac{N}{f_t}\right)$
IDF_B2	Logarithmic formulation used in *ltc* variant	$w_t = \log_e \left(\frac{N}{f_t}\right)$
IDF_C	Hyperbolic formulation	$w_t = \frac{1}{f_t}$
IDF_D	Normalized formulation	$w_t = \log_e \left(1 + \frac{f_m}{f_t}\right)$
IDF_E	Another normalized formulation SB = p	$w_t = \log_e \frac{N - f_t}{f_t}$
	The following definitions are based on the term's noise n_t and signal s_t.	$n_t = \sum_{d \in \mathcal{D}_t} \left(-\frac{f_{d,t}}{F_t} \log_2 \frac{f_{d,t}}{F_t}\right)$ $s_t = \log_2(F_t - n_t)$
IDF_F	Signal	$w_t = s_t$
IDF_G	Signal-to-Noise ratio	$w_t = \frac{s_t}{n_t}$
IDF_H		$w_t = \left(\max_{t' \in \mathcal{T}} n_{t'}\right) - n_t$
IDF_I	Entropy measure	$w_t = 1 - \frac{n_t}{\log_2 N}$
IDF_J	Okapi BM25 IDF formulation, according to [18, 19]	$w_t = \log \frac{N - f_t + 0.5}{f_t + 0.5}$

Table 4. Evaluated variants to calculate the inverse document frequency w_t.

Abbr.	Description	Formulation
SIM_INN	Inner Product	$S_{d_1,d_2} = \sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} \cdot w_{d_2,t})$
SIM_COS	Cosine Measure	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} \cdot w_{d_2,t})}{W_{d_1} \cdot W_{d_2}}$
SIM_DIC	Dice Formulation	$S_{d_1,d_2} = \frac{2 \sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} \cdot w_{d_2,t})}{W_{d_1}^2 + W_{d_2}^2}$
SIM_JAC	Jaccard Formulation	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} \cdot w_{d_2,t})}{W_{d_1}^2 + W_{d_2}^2 - \sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} \cdot w_{d_2,t})}$
SIM_OVL	Overlap Formulation	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} \cdot w_{d_2,t})}{\min(W_{d_1}^2, W_{d_2}^2)}$
SIM_EUC	Euclidean Similarity	$D_{d_1,d_2} = \sqrt{\sum_{t \in \mathcal{T}_{d_1,d_2}} (w_{d_1,t} - w_{d_2,t})^2}$ $S_{d_1,d_2} = \left(\max_{d'_1,d'_2}(D_{d'_1,d'_2})\right) - D_{d_1,d_2}$
SIM_JEF	Jeffrey Divergence-based Similarity	$S_{d_1,d_2} = \left(\max_{d'_1,d'_2}(D_{d'_1,d'_2})\right) - D_{d_1,d_2}$ $D(F,G) = \sum_i \left(f_i \log \frac{f_i}{m_i} + g_i \log \frac{g_i}{m_i}\right)$ $m_i = \frac{f_i + g_i}{2}$

Table 5. Evaluated similarity functions S_{d_1,d_2}.

3. EVALUATION

We performed *genre classification* experiments to evaluate the different algorithmic choices discussed in the previous section. Although genre taxonomies are often inconsistent and erroneous [15], it is commonplace in MIR to use genre as a proxy for artist similarity. The evaluated retrieval task consists of determining k artists similar to a given query artist. This task resembles k nearest neighbor classification, where the genre of a seed artist is predicted as the most frequent genre among the seed's k most similar artists.

3.1 Data Sets

We used two data sets for evaluation. The first one, referred to as C224a, consists of 224 well-known artists and has a uniform genre distribution (14 genres [1], 16 artists each). It has been frequently used to evaluate Web-/text-based MIR approaches.

The second data set C3ka consists of 3,000 music artists, representing a real-world collection. The data has been gathered as follows. We used last.fm's API to extract the most popular artists for each country of the world, which we then aggregated into a single list. Since last.fm's data is prone to misspellings due to its collaborative nature, we cleaned the data set by matching each artist name with the database of the expert-based music information system allmusic.com, from which we also extracted genre information. Starting this matching process from the most popular artist found by last.fm and including only names that also occur in allmusic.com, we eventually obtained a list of 20,995 artists, out of which we selected the top 3,000. These artists are categorized into 18 distinct genres [2] according to allmusic.com. Both data sets are available for download. [3]

3.2 Experiments

To gather music-related posts, we use Twitter's API. Accounting for the time-varying behavior of the search results and to obtain a broad coverage, we queried Twitter during February/March 2010 and December 2010/January 2011, yielding a total of about six million tweets. For artist set C224a, we achieved a coverage of 100%; for set C3ka, we achieved a coverage of 96.87%.

We employed a two-staged evaluation, similar to [22]: In order to filter inferior algorithmic combinations, we first investigated each algorithmic setting on data set C224a. [4] In a second set of experiments, we then evaluated the remaining variants on the real-world artist set C3ka. As performance measure *Mean Average Precision* (MAP) is used.
In the first stage of the experiments, only variants that fulfill at least one of the following two conditions are retained:

- there is a relative MAP difference of 10% or less to the top-ranked variant
- or the t-test does not show a significant difference to the top-ranked variant (at 5% significance level).

The top 577 variants have a relative MAP difference of less than 10% to the highest ranked combination. The pairwise t-test shows a significant difference for the top-ranked 1,809 variants. For the second stage of experimentation, conducted

[1] The genres in C224a are Country, Folk, Jazz, Blues, R'n'B/Soul, Heavy Metal/Hard Rock, Punk, Rap/Hip Hop, Electronica, Reggae, Rock'n'Roll, Pop, and Classical.

[2] The genres in C3ka are Avantgarde, Blues, Celtic, Classical, Country, Easy Listening, Electronica, Folk, Gospel, Jazz, Latin, Newage, Rap, Reggae, RnB, Rock, Vocal, and World.

[3] http://www.cp.jku.at/people/schedl/datasets.html

[4] Excluding redundant combinations, a total of 23,100 single experiments have been conducted in this stage.

on collection C3ka, we therefore evaluated only these top-ranked 1,809 variants.

3.3 Results and Discussion

Table 6 shows the 10 top-ranked and the 10 bottom-ranked variants with their MAP scores (considering 15 nearest neighbors) for set C224a. The MAP scores of the 23,100 evaluated variants span a wide range and are quite diverse, with a mean of $\mu = 37.89$ and a standard deviation of $\sigma = 17.16$. From Table 6 it can be seen that highest MAP scores are achieved when using QS_A, TS_A, and NORM_NO. At the other end of the ranking we see that QS_M and SIM_OVL dominate the most inferior variants.

To obtain a better understanding of the individual components that contribute to a well-performing social similarity measure, we analyzed the distribution of each aspect among the 1,809 top-ranked variants:

Regarding the query scheme, using only the artist name as indicator to determine related tweets (QS_A) outperforms adding music-specific key words. It seems that additional key words too heftily prune Twitter's result set.

As for the term sets used for indexing, the top ranks are dominated by algorithmic variants that use the whole set of terms (TS_A). It is noteworthy, however, that the good performance of TS_A and TS_S comes at the price of much higher computational complexity (cf. Table 2). Hence, when performance is crucial, the results suggest using other term sets. A particularly good choice seems to be TS_N, the list of artist names, as it is the set that most frequently occurs among the top-ranked variants (32.5%). Another interesting finding is that the music dictionary TS_D, despite its good performance for artist clustering based on *Web pages*, cf. [16], occurs first only at rank 1,112. An empirically verified reason for this may be that Twitter users tend to refrain from using a comprehensive music-specific vocabulary, even when they twit about music-related issues. [5]

As for the term weighting functions (TF and IDF variants), no clear picture regarding favorable variants emerges from the experiments. We found, however, that TF_A only occurs in 3.15% of the top-ranked variants and should thus be avoided. The most frequently occurring formulations on the other hand are TF_C2 (15.69%) and TF_E (16.80%), the latter being particularly present in the very top ranks. Analogous to TF, for IDF variants we can easily point to formulations that should be avoided, namely IDF_G (0.50% occurrence), IDF_F (0.66%), and IDF_A (2.54%). The IDF variants most frequently occurring within the top ranks are IDF_B2 (13.93%), IDF_J (13.71%), and IDF_E (13.38%).

As for the similarity measure, we found no clear evidence that cosine similarity (SIM_COS), the de-facto standard measure in IR, generally outperforms the others. It is likely that the key advantage of SIM_COS, the document length normalization, plays a minor role, because tweets are limited to 140 characters which are usually exhausted. Further support for this hypothesis is given by the remarkably good performance of the simple inner product measure (SIM_INN) that does not perform any length normalization. Also among the virtual document normalization methods, using no normalization at all (NORM_NO) outperformed the other variants investigated, accounting for 52.24% of the top ranks.

[5] Only 478 unique terms out of the 1,398 in TS_D were used, only 319 were used in at least two different tweets.

MAP	Variant
64.018	QS_A.TS_A.NORM_NO.TF_C2.IDF_E.SIM_JAC
63.929	QS_A.TS_A.NORM_NO.TF_C2.IDF_J.SIM_JAC
63.839	QS_A.TS_A.NORM_NO.TF_C.IDF_E.SIM_JAC
63.810	QS_A.TS_A.NORM_NO.TF_C2.IDF_E.SIM_COS
63.780	QS_A.TS_A.NORM_NO.TF_C.IDF_E.SIM_COS
63.780	QS_A.TS_A.NORM_NO.TF_C2.IDF_B2.SIM_JAC
63.780	QS_A.TS_A.NORM_NO.TF_C2.IDF_B2.SIM_DIC
63.720	QS_A.TS_A.NORM_NO.TF_C2.IDF_E.SIM_DIC
63.601	QS_A.TS_A.NORM_NO.TF_C2.IDF_J.SIM_COS
63.542	QS_A.TS_A.NORM_NO.TF_C.IDF_J.SIM_JAC
...	...
3.482	QS_M.TS_A.NORM_MAX.TF_G.IDF_G.SIM_OVL
3.452	QS_M.TS_S.NORM_SUM.TF_B.IDF_F.SIM_OVL
3.423	QS_M.TS_A.NORM_SUM.TF_C3.IDF_J.SIM_OVL
3.363	QS_M.TS_S.NORM_MAX.TF_G.IDF_F.SIM_OVL
3.274	QS_M.TS_A.NORM_SUM.TF_C.IDF_E.SIM_OVL
3.065	QS_M.TS_A.NORM_SUM.TF_C.IDF_J.SIM_OVL
3.006	QS_M.TS_A.NORM_MAX.TF_G.IDF_F.SIM_OVL
2.976	QS_M.TS_S.NORM_MAX.TF_F.IDF_F.SIM_OVL
2.857	QS_M.TS_A.NORM_MAX.TF_F.IDF_G.SIM_OVL
2.649	QS_M.TS_A.NORM_MAX.TF_F.IDF_F.SIM_OVL

Table 6. MAP scores of the top-ranked and bottom-ranked variants on set C224a.

MAP	Variant
72.570	QS_A.TS_S.NORM_NO.TF_G.IDF_H.SIM_JAC
72.566	QS_A.TS_S.NORM_NO.TF_G.IDF_H.SIM_DIC
72.553	QS_A.TS_S.NORM_NO.TF_C.IDF_E.SIM_COS
72.553	QS_A.TS_S.NORM_NO.TF_C.IDF_J.SIM_COS
72.536	QS_A.TS_S.NORM_NO.TF_F.IDF_H.SIM_DIC

Table 7. MAP scores of the top 5 variants on set C3ka.

On the second data set, C3ka, the achieved results were comparable. Spearman's rank-order correlation coefficient computed on the two rankings obtained with the two artist sets revealed a moderate correlation of 0.37. This indicates that the rankings produced by the same algorithmic choices are not largely influenced by factors such as size of artist collection or number of artists per genre. Table 7 contains the five top-ranked variants for set C3ka.

4. CONCLUSIONS AND OUTLOOK

We presented a large-scale evaluation of using Twitter posts for the purpose of artist similarity estimation. To this end, we analyzed 23,100 algorithmic choices related to query scheme, index term set, length normalization, term weighting function, and similarity measure, using two data sets of music artists. The main findings can be summarized as follows:

- Restricting the search by additional key words prunes the resulting set of tweets too heavily. Using only the artist name as query (QS_A) should be favored.
- Best results are achieved using all terms in the corpus (TS_A), though at high computational costs. When computational complexity is an issue, the results suggest using artist names as index term set (TS_N).

- Normalizing for length does not significantly improve the results, neither on term vectors, nor in the similarity function. Taking into account the higher computational costs, we therefore recommend refraining from normalization (`NORM_NO`) and using, for example, the inner product as similarity measure (`SIM_INN`).
- The simple binary match TF formulation `TF_A` should not be used. The most favorable variants are `TF_C2` and in particular `TF_E`.
- Among the IDF formulations, we suggest to refrain from using `IDF_A`, `IDF_F`, and `IDF_G`. Better alternatives are given by formulations `IDF_B2`, `IDF_E`, and `IDF_J`.

Future work will focus on investigating the performance of different approaches on the "long tail" of artists and on incorporating temporal and geographic properties of tweets. The contextual similarity measures analyzed in this work will help develop more accurate social and personalized models of musical similarity. Combined with content-based models, they might pave the way for a new generation of personalized music applications, such as intelligent recommenders or playlist generators.

5. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Funds (FWF): P22856-N23 and L511-N15. We further wish to thank *Tim Pohle* for his contributions.

6. REFERENCES

[1] J.-J. Aucouturier and F. Pachet. Scaling Up Music Playlist Generation. In *Proc. IEEE ICME*, Aug 2002.

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

[3] S. Baumann and O. Hummel. Using Cultural Metadata for Artist Recommendation. In *Proc. WEDELMUSIC*, Sep 2003.

[4] Ò. Celma. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[5] http://www.cp.jku.at/CoMIRVA (access: Jan 2011).

[6] F. Debole and F. Sebastiani. Supervised Term Weighting for Automated Text Categorization. In *Proc. ACM SAC*, Mar 2003.

[7] M. Evans. Twitter Enjoys Major Growth and Excellent Stickiness. http://blog.sysomos.com (access: Jan 2011).

[8] http://www.freebase.com (access: Jan 2011).

[9] X. Hu and J.S. Downie. Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata. In *Proc. ISMIR*, Sep 2007.

[10] P. Knees, E. Pampalk, and G. Widmer. Artist Classification with Web-based Data. In *Proc. ISMIR*, Oct 2004.

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

[12] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung. A Comprehensive Comparative Study on Term Weighting Schemes for Text Categorization with Support Vector Machines. In *Proc. ACM WWW*, May 2005.

[13] J. Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Trans. Information Theory*, 37, 1991.

[14] http://lucene.apache.org (access: Jan 2011).

[15] F. Pachet and D. Cazaly. A Taxonomy of Musical Genre. In *Proc. RIAO*, Apr 2000.

[16] E. Pampalk, A. Flexer, and G. Widmer. Hierarchical Organization and Description of Music Collections at the Artist Level. In *Proc. ECDL*, Sep 2005.

[17] E. Pampalk and M. Goto. MusicSun: A New Approach to Artist Recommendation. In *Proc. ISMIR*, Sep 2007.

[18] J. Pérez-Iglesias, J. R. Pérez-Agüera, V. Fresno Y. Z., and Feinstein. Integrating the Probabilistic Models BM25/BM25F into Lucene. *CoRR*, 2009.

[19] S.E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive Track. In *Proc. TREC*, 1999.

[20] G. Salton and C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5), 1988.

[21] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11), 1975.

[22] M. Sanderson and J. Zobel. Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. In *Proc. ACM SIGIR*, Aug 2005.

[23] M. Schedl. *Automatically Extracting, Analyzing, and Visualizing Information on Music Artists from the World Wide Web*. PhD thesis, JKU Linz, Austria, 2008.

[24] M. Schedl. On the Use of Microblogging Posts for Similarity Estimation and Artist Labeling. In *Proc. ISMIR*, Aug 2010.

[25] M. Schedl, P. Knees, K. Seyerlehner, and T. Pohle. The CoMIRVA Toolkit for Visualizing Music-Related Data. In *Proc. of the 9th Eurographics/IEEE VGTC Symposium on Visualization (EuroVis 2007)*, May 2007.

[26] M. Schedl, P. Knees, and G. Widmer. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proc. CBMI*, Jun 2005.

[27] M. Schedl, T. Pohle, P. Knees, and G. Widmer. Exploring the Music Similarity Space on the Web. *ACM Transactions on Information Systems*, 29(3), July 2011.

[28] http://wordlist.sourceforge.net (access: Jan 2011).

[29] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards Musical Query-by-Semantic Description using the CAL500 Data Set. In *Proc. ACM SIGIR*, Jul 2007.

[30] B. Whitman and S. Lawrence. Inferring Descriptions and Similarity for Music from Community Metadata. In *Proc. ICMC*, Sep 2002.

[31] J. Yarow. Twitter Finally Reveals All Its Secret Stats. http://www.businessinsider.com (access: Jan 2011).

[32] J. Zobel and A. Moffat. Exploring the Similarity Space. *ACM SIGIR Forum*, 32(1), 1998.

[33] J. Zobel and A. Moffat. Inverted Files for Text Search Engines. *ACM Computing Surveys*, 38, 2006.