Multimedia Systems (2014) 20:693–705
DOI 10.1007/s00530-013-0321-5

REGULAR PAPER

Harvesting microblogs for contextual music similarity estimation: a co-occurrence-based framework

Markus Schedl · David Hauger · Julián Urbano

Published online: 9 May 2013

Abstract Microtexts are a valuable, albeit noisy, source to infer collaborative information. As music plays an important role in many human lives, microblogs on music-related activities are available in abundance. This paper investigates different strategies to estimate music similarity from these data sources. In particular, we first present a framework to extract co-occurrence scores between music artists from microblogs and then investigate 12 similarity estimation functions to subsequently derive resemblance scores. We evaluate the approaches on a collection of microblogs crawled from `Twitter` over a period of 10 months and compare them to standard *tf-idf* approaches. As evaluation criteria we use *precision* and *recall* in an artist retrieval task as well as *rank proximity*. We show that collaborative chatter on music can be effectively used to develop *music artist similarity measures*, which are a core part of every music retrieval and recommendation system. Furthermore, we analyze the effects of the "long tail" on retrieval results and investigate whether results are consistent over time, using a second dataset.

This work was supported by the Austrian Science Funds (FWF): P22856 and P25655, and by the FP7-ICT-2011-9 project: 601166.

M. Schedl (✉) · D. Hauger
Department of Computational Perception, Johannes Kepler University, Linz, Austria
e-mail: Markus.Schedl@jku.at

D. Hauger
e-mail: david.hauger@jku.at

J. Urbano
Department of Computer Science, University Carlos III of Madrid, Madrid, Spain
e-mail: jurbano@inf.uc3m.es

Keywords Social media mining · Music information retrieval · Microblog analysis · Similarity measurement · Trend prediction

1 Motivation

Developing music similarity measures that reflect resemblance perceived by humans is one of the big challenges in music information retrieval (MIR), a subfield of multimedia information retrieval. These similarity measures enable applications such as music recommender systems [4, 8], automated playlist generators [20, 25], or intelligent user interfaces to music collections [23, 19]. Computational features for music similarity calculation can be broadly categorized into *music content-based*, *music context-based*, and *user context-based* [34]. While content-based feature extraction techniques derive the representation of a music item from the audio signal itself [7], music context-based approaches make use of data that are not encoded in the audio signal [30], for instance, the performer's political background, the meaning of a song's lyrics, images of album covers, or co-occurrence information derived from playlists.

Representing a rich source of user-generated content, microblogs are well suited to derive such music context-based similarity features [31]. In fact, today's most popular microblogging service, `Twitter`[1], has more than 200 million registered users[2] who are creating a billion posts every week[3] (as of March/April 2011). As music plays an

[1] http://www.twitter.com.

[2] http://huffingtonpost.com/2011/04/28/twitter-number-of-users_n_855177.html.

[3] http://blog.twitter.com/2011/03/numbers.html.

important role in many human lives, it is an omnipresent topic on the (social) web. Almost everybody enjoys listening to his favorite tunes, and many people share their opinions about songs, artists, or latest album releases. Some even share their own versions of favored music videos. Digital music distribution and consumption are also important economic factors, which is demonstrated by the current success of music streaming services such as `Spotify`.[4]

The work at hand, as far as we are aware of, is one of the first to thoroughly evaluate different strategies to mine the microblogosphere to infer music similarity. The remainder of the paper is organized as follows. Section 2 reviews related literature on text-based music similarity measurement and social media retrieval. Section 3 reports on the acquisition of music-related tweets and presents results of statistical data analyses. Our framework to infer similarity between music artists from microblogs is presented in Sect. 4. In Sect. 5 we evaluate and analyze the proposed co-occurrence approaches and compare them to standard $tf \cdot idf$-based algorithms. We further analyze temporal stability and influence of artist popularity on retrieval results. Eventually, Sect. 6 summarizes the main findings and points to future research directions.

2 Related work

The work at hand is strongly related to other *text-based approaches to music similarity measurement* and to *social media retrieval*. Literature on both research areas are discussed in the following.

Although *modeling text documents* using vector space representations has a long tradition in IR research [3, 27], similar models targeted at music and multimedia retrieval did not emerge until about a decade ago. Indeed, deriving *term feature vectors from web pages* for the purpose of music artist similarity estimation was first proposed in 2000 by Cohen and Fan [12]. They extract lists of artist names from web pages determined by querying web search engines. The resulting pages are then parsed according to their DOM tree, filtered, and sought for occurrences of entity names. Term vectors of co-occurring artist names are subsequently used for artist recommendation. Using artist names to build term vector representations, whose term weights are computed as co-occurrence scores, is an approach also followed later in [14, 36, 41]. In contrast to Cohen and Fan's approach, Zadel and Fujinaga [41] and Schedl et al. [36] derive the term weights from *search engine's page count estimates* and suggest their method for artist recommendation. Automatically querying a web search engine to determine pages related to a specific topic is a common and intuitive strategy, which is therefore frequently performed for data acquisition in information extraction tasks. Examples in the music domain can be found in [13, 40], whereas [10, 11, 18] apply this technique in a more general context.

Computing term feature vectors from term sets other than artist names is performed by Whitman and Lawrence [40]. They extract different term sets (unigrams, bigrams, noun phrases, artist names, and adjectives) from up to 50 artist-related web pages obtained via a search engine. After downloading the pages, the authors apply parsers and a part-of-speech (POS) tagger to assign each word to its suited test set(s). An individual term profile for each artist is then created by employing $tf \cdot idf$ weighting. The overlap between the term profiles of two artists, i.e., the sum of weights of all terms that occur in both term profiles, is then used as an estimate of their similarity. Extending the work presented in [40], Baumann and Hummel [5] introduce various filters to prune the set of retrieved web pages (length-based filtering, advertisement filtering, and keyword spotting in the URL, the title, and the first text part of each page).

Unlike Whitman and Lawrence, who study with different term sets, Knees et al. [17] present a similar approach using only one list of unigrams. For each artist, a weighted term profile is created by applying a $tf \cdot idf$ variant, and cosine similarity is used to compute resemblance between these term profiles. The authors evaluate their approach in a genre classification setting using as classifiers k-nearest neighbor (kNN) and support vector machines (SVM) [38]. Govaerts et al. [14] evaluate a similar approach, particularly focusing on temporal and regional differences between search engines.

Other approaches derive term profiles from more specific web resources. Celma et al. [9] propose a music search engine that crawls *audio blogs via RSS feeds* and calculates $tf \cdot idf$ features. Hu et al. [15] extract *tf*-based features from music reviews gathered from `Epinions.com`.[5] Schedl [28] extracts user posts from `Twitter` associated with music artists and models term profiles using term lists specific to the music domain. Although one of the goals (artist similarity measurement) and the data source (microblogs) resemble the work at hand [28] bases the similarity computation on $tf \cdot idf$ representations of music artists, whereas the approaches reported in this paper derive a similarity estimate from co-occurrence information. Schedl [31] presents a more general framework to derive $tf \cdot idf$-based similarity measures from microblogs in an effort to estimate similarities between music artists and between movies. To this end, thorough evaluation

[4] http://www.spotify.com.

[5] http://www.epinions.com/music.

experiments have been conducted to analyze various aspects of the term vector space model: query scheme, index term sets, term frequency, inverse document frequency, similarity measure, and normalization approaches.

In contrast to our earlier work [28, 31], the paper at hand defines a framework for music artist similarity estimation based on *co-occurrences* of artist names among `Twitter` users. We will show in Sect. 5 that this co-occurrence based approach outperforms the method of employing $tf \cdot idf$-like weighting schemes. In addition, here we investigate the temporal stability of retrieval quality, which is particularly important for a data source like microblogs, where content changes instantaneously. We further address the question whether the popularity of an artist influences the similarity estimates.

The second related research area, namely that of *social media retrieval*, is a research field that covers all aspects of information retrieval in the context of social media. A good literature overview about recent trends in this emerging area can be found in [26].

As for *document modeling in social media*, particularly in microblogs, it is shown by Metzler et al. [21] that $tf \cdot idf$ models, although being the standard term weighting approach in Text-IR, seems rather unsuited to model microblogs. Further support for this finding is given by Naveed et al. [22] who argue that $tf \cdot idf$ is a poor metric for term weighting in microblog retrieval tasks due to the term sparsity in microblogs and the frequently poor quality of texts. As for the former, the very restricted length of microblogs (140 characters in the case of `Twitter`) leads to retrieval problems, because the limited number of terms in a microblog post harms retrieval performance if the query does not contain one of the few terms. Also, the motivation for document length normalization is no longer valid when dealing with tweets whose length is usually just a bit below the maximum of 140 characters. Microblogs further suffer from very diverse content quality, which also affects retrieval performance. In a music retrieval scenario, Schedl presents in [31] the consistent finding that document length normalization of microblogs does not improve retrieval performance. Likewise, the cosine similarity measure does not perform better than the simple inner product.

Particularly focusing on *social media music retrieval*, Schedl proposes in [32] a standardized corpus of data on music listening behavior mined form microblogs. The paper further reports findings of correlation analyses investigating the spatial and temporal stability of the listening patterns. It is shown that listening patterns are independent of the month, but highly dependent on the day of the week (workdays vs. weekends) and on the country.

There also exists some related work on *user retrieval and recommendation* in the microblogosphere. For instance, Armentano et al. [1] present a recommender system that suggests users to follow based on tweet similarity of microbloggers. To this end, the authors create and investigate different user profiles, for example, modeling the seed user via term frequencies of his/her aggregate posts or of all of his/her followees. In a related work, Weng et al. [39] aim at identifying influential `Twitter` user for a given topic. To this end, they apply latent Dirichlet allocation (LDA) [6] to their corpus of tweets. Subsequently, topical similarity between users is computed as the Jensen–Shannon divergence between the distribution of the latent topics of the respective users. Further taking into account the link structure, Weng et al. propose a ranking function for influential users in each topic. Similar to Armentano et al., they evaluate their approach in a recommendation setting.

Finally, although not closely related to social media retrieval, the work by Peat and Willet [24] is nevertheless important for the article at hand. The authors investigate the merits and limitations of using *term co-occurrences to model term similarity*. Even though they focus on query expansion, their main findings also influence our work. Peat and Willet show that (1) terms with comparable term frequencies in the corpus are usually more similar to each other than terms with different frequencies of occurrence and (2) those with high occurrence figures are poor discriminators between relevant and non-relevant documents. These findings strongly support the integration of a popularity correction factor into the proposed similarity models, as suggested in Sect. 4. Its purpose is to alleviate distortions in similarity measures that are caused by very popular artists who are listened to by almost everyone.

3 Data acquisition and analysis

We crawled `Twitter` postings containing the hashtag `#nowplaying` between May 2010 and March 2011, as this hashtag has already proven successful to determine music listening-related tweets [29]. The crawls were restricted to tweets with geospatial information and to all cities with more than 500,000 inhabitants (790 cities around the world were gathered from `World Gazetteer`[6]). Between November 2010 and March 2011, we gathered another dataset, focusing on tweets containing the hashtag `#itunes`, because it is frequently used among users of `Apple`'s `iTunes` and related programs. There also exists a popular plug-in for `Apple`'s social network `Ping` that automatically tweets `iTunes` listening activities using this very hashtag.

[6] http://www.world-gazetteer.com.

We were able to retrieve 9,928,817 tweets for #nowplaying (686,867 users) and 725,486 tweets for #itunes (91,768 users). We will henceforth refer to these datasets simply as #nowplaying and #itunes. To use the crawled tweets for artist similarity estimation, we had to map them to artist names. Standard $tf \cdot idf$ approaches may result in strong biases when dealing with microblogs, as they—by definition—only consist of a small number of words [22]. Therefore, we have been striving for alternative approaches. Despite including comments, most tweets on music listening are similarly structured, which becomes even more evident if these tweets are automatically generated. To this end, we identified a number of common patterns in the tweets, such as:

- *song title* by *artist name* [on *some platform*]
- *artist name*: "*song title*"
- *song title #artist name*
- *song title – artist name*
- *artist name – song title*

To give an impression on how such potentially music-related tweets containing the desired hashtags look like, here are some examples from our dataset, including tweets that do not contain track information:

1. likes Go by Hillsong United on Ping http://t.co/n2w9nEv #iTunes
2. #Nowplaying : Yesterday ~ The Beatles
3. "That boy is a monster, m-m-m-monster..." #nowplaying Monster by Lady Gaga
4. le gusta Ultimate Tracks: Does Anybody Hear Her (As Made Popular by Casting Crowns) [Performance Track] de ... http://t.co/8gtRYIV #
5. #NowPlaying 2 in a row from Adam Lambert with a special acoustic version of Sleepwalker recorded at SWR3 Germany
6. created the playlist christmas time http://t.co/h6kEUrQ #iTunes
7. #nowplaying street soccer
8. 7 min for #iTunes announcement. Have been waiting all day.. Hope it is something really good

We subsequently matched the potential artist names against a list of 455,087 artists and 7,795,612 tracks from the musicbrainz[7] database. We implemented a multi-level system that tries to apply specific and restrictive patterns first (e.g., "likes *song title* by *artist name* on *some platform*"—see example 1) and proceeds with more general ones (e.g., terms separated by special characters—see example 2). Many tweets contain comments (see examples 3 and 4) and are still correctly analyzed by our system. Sometimes tweets are related to music, but without containing explicit track information (see examples 5 and 6). These tweets are not mapped to specific tracks—future work might use link following to obtain additional information. As there are only conventions but no restrictive rules on the use of hashtags, some tweets might not be related to music listening behavior (see examples 7 and 8).

Employing this artist detection approach, we were able to identify 38,183 unique artists in 2,945,780 (29.7 %) of the tweets containing #nowplaying. Each artist appeared between 1 and 9,066 ("Paramore") times (*mean* = 24.07, *std.dev.* = 177.02, *median* = 2). From the #itunes dataset, we extracted 11,804 artists from 198,185 (27.3 %) tweets. The most frequently occurring band was "The Beatles" with 939 individual tweets (*mean* = 5.96, *std.dev.* = 21.46, *median* = 1), which may be related to the fact that in November 2011 itunes started providing their songs. A complete list of the top 20 artists for both hashtags is given in Table 5.

As for geographical coverage, tweets from 766 (97 %) different cities in 127 countries were retrieved using #nowplaying. The #itunes collection covers 603 (76 %) cities in 107 countries. Tables 1 and 2 show the top ten cities and countries, respectively, in terms of the number of postings. From these tables, we can already see that Apple products are particularly widespread in the USA as the number of tweets including #itunes is higher in the USA than in any other country, not only in absolute numbers but also relative to the respective population size.

Figure 1 visualizes on log–log scale the relative distributions of play counts for datasets #nowplaying and #itunes. It can be seen that both distributions are indeed very similar for the most part. However, the #itunes distribution has more mass under the most popular artists and also under unpopular artists. This evidences a clear bias in the #itunes dataset: the most popular artists (left side) are even more popular, relative to the others, than in

Table 1 Top ten cities (per number of tweets) in both datasets

#nowplaying		#itunes	
City	Tweets	City	Tweets
New York	126,952	New York	13,603
London	96,801	London	9,813
São Paulo	79,317	Los Angeles	9,030
Los Angeles	73,834	San Francisco	5,787
Amsterdam	66,021	San Jose	5,605
Guarulhos	58,453	Chicago	4,413
Osasco	57,512	Birmingham	3,869
São Bernardo	56,946	Toronto	3,363
Rotterdam	55,113	Hamilton	3,279
Mexico City	52,618	Baltimore	3,245

[7] http://www.musicbrainz.org.

Table 2 Top ten countries (per number of tweets) in both datasets

#nowplaying		#itunes	
Country	Tweets	Country	Tweets
Brazil	725,389	USA	78,460
USA	673,839	Japan	30,932
Japan	458,558	Mexico	23,047
Mexico	419,584	Brazil	16,390
Indonesia	284,082	UK	15,134
South Korea	251,132	Canada	11,266
China	183,178	South Korea	8,652
UK	128,744	Australia	5,119
Netherlands	121,134	China	4,492
Venezuela	110,336	Germany	3,157

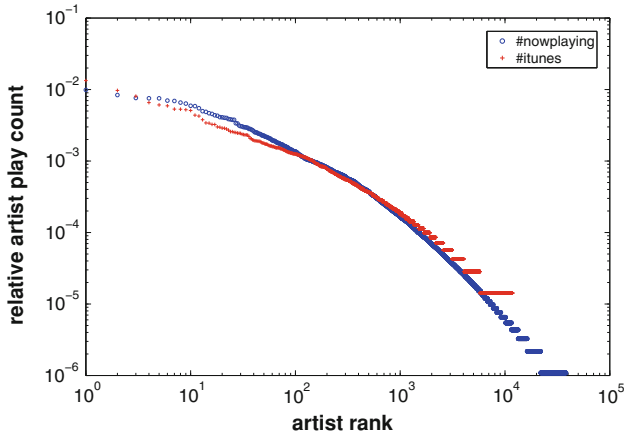

Fig. 1 Distribution of relative artist play counts for #nowplaying and #itunes

the #nowplaying dataset. Likewise, the most unpopular artists (right side) are also more popular in the #itunes dataset, showing a larger lack of diversity and thus a shorter "long tail" than in #nowplaying. The reason for these differences can be found in the sampling process. The #itunes hashtag represents users of iTunes for the most part, while the #nowplaying hashtag is more general and therefore represents a wider population of users. For a comparison of listening habits expressed by #itunes and by #nowplaying, the interested reader is referred to [33].

To further analyze similarities between datasets, we also compared the relative play counts on a per-artist basis. This way, we can test whether the high similarity in the distributions is actually due to the same artists appearing with similar relative counts in both datasets. Figure 2 shows how the relative play count in one dataset fits with the play count in the other, each point representing one of the 9,813 artists that appear in both datasets. The straight red line

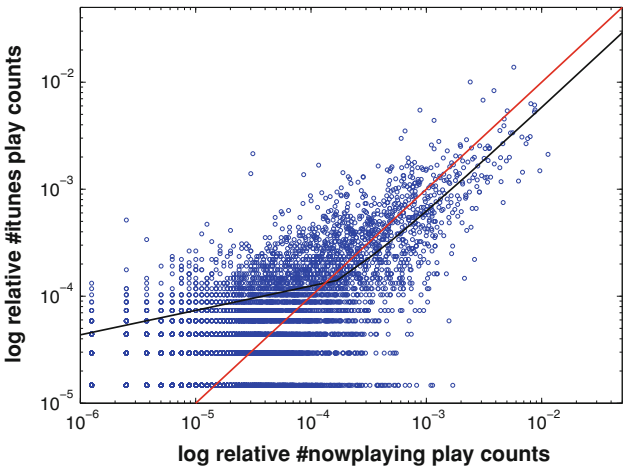

Fig. 2 Relative artist play counts for #nowplaying versus #itunes

represents points with same play counts; the black line indicates a linear regression of the data. Please note that both axes are again logarithmically scaled. Indeed, we can see a very clear correlation, though highly sparse (Spearman's $\rho = 0.659$). Most importantly, we can see that the fit is better with high play counts, meaning that popular artists are indeed popular in both datasets. With unpopular artists, there is too much variation, showing the usual diversity in the long tail.

To summarize, both datasets show similar distributions, but #itunes seems to have a clear bias both in terms of underlying users and relative popularity of top artists, as well as a lower degree of diversity. In addition, it is an order of magnitude smaller than #nowplaying. Therefore, and unless otherwise indicated, in the remaining parts of the paper we use the #nowplaying dataset.

4 A framework for music similarity estimation based on co-occurrences

We define a family of similarity estimation functions between arbitrary music artists i and j defined as the product

$$sim(i,j) = w(i,j) \cdot (i,j)$$

where $w(i, j)$ is a scoring function and $p(i, j)$ is an optional popularity correction factor. We examine six different scoring functions:

$$w_1(i,j) = \frac{cooc_{i,j}}{occ_i} \quad (1)$$

$$w_2(i,j) = \frac{cooc_{i,j}}{\min(occ_i, occ_j)} \quad (2)$$

$$w_3(i,j) = \frac{cooc_{i,j}}{\max(occ_i, occ_j)} \quad (3)$$

$$w_4(i,j) = \frac{cooc_{i,j}}{\frac{1}{2} \cdot (occ_i + occ_j)} \quad (4)$$

$$w_5(i,j) = \frac{cooc_{i,j}}{occ_i \cdot occ_j} \quad (5)$$

$$w_6(i,j) = \frac{cooc_{i,j}}{\sqrt{occ_i \cdot occ_j}} \quad (6)$$

where $cooc_{i,j}$ denotes the co-occurrence count of artists i and j on a per-user basis (i.e., the number of users who listen to both artists i and j), and occ_i is the total number of users who listen to artist i. Resembling the proposal by Whitman and Lawrence [40], we define the popularity correction factor as

$$p(i,j) = 1 - \frac{|occ_i - occ_j|}{\max_k occ_k}$$

where $\max_k occ_k$ denotes the maximum frequency of an artist k in the whole set of artists. This term aims at alleviating the popularity bias, i.e., distortions in similarity estimates caused by very popular artists, which are found in almost every music collection. Hence, similarity scores between artists of highly differing popularity are downweighted. In variants without popularity correction, we set $p(i, j) = 1$. Among these similarity functions, $w_1(i, j)$ may be regarded as baseline. Accounting for one artist only, it is asymmetric and therefore does not reflect the similarity between two artists, but represents the relative frequency. It is used as approximation for the conditional probability of users listening to artist j, given that they are listening to artist i.

We denote variants of the similarity function as s or *spop*, where s identifies the scoring function (Eqs. 1–6) and *pop* denotes the use of popularity correction. For instance, *1pop* refers to the similarity measure $\frac{cooc_{i,j}}{occ_i} \cdot \left(1 - \frac{|occ_i - occ_j|}{\max_k occ_k}\right)$, while *3* refers to the measure $\frac{cooc_{i,j}}{\max(occ_i, occ_j)}$. In total, we thus investigate 12 similarity estimation functions.

Since *tf* · *idf* weighting is the standard approach in Text-IR, we investigate its performance for similar artist retrieval on microblog data as a baseline for our co-occurrence framework. Following the suggestion by Schedl et al. [35], who present a large-scale study on *tf* · *idf*-based similarity computation algorithms on microblogs, we compute the weight of a term t in a document d (which is a concatenation of all tweets retrieved for the artist under consideration) as

$$w_{d,t} = \ln(1 + f_{d,t}) \cdot \ln \frac{N - f_t}{f_t}$$

where $f_{d,t}$ is the frequency of term t in document d, N is the total number of artists, and f_t is the number of artist documents where term t occurs. As similarity measure between the *tf* · *idf* vectors, we use the cosine similarity. This version of the *tf* · *idf* model proved particularly beneficial for modeling pre-filtered music-related microblogs [31]. We further analyze different term sets on which the *tf* · *idf* weights are calculated. The set denoted "lastfm" comprises the 250 tags most frequently assigned by users of `last.fm`[8], whereas set "dict" is a dictionary of 1,379 music terms, provided by the authors of [35].

5 Evaluation

Music retrieval algorithms are traditionally evaluated using genre information as proxy for similarity and performing genre classification experiments. However, this approach is ambivalently discussed in the MIR community [2], some of the reasons being that genre is a fuzzily defined concept, different genres overlap, subjective interpretations of one and the same genre often vary, and it is not at all clear how genres relate to musical aspects perceived by humans.

5.1 Similar artist retrieval evaluated on collaborative data

To avoid these issues, we opt for an evaluation strategy that compares the algorithms' output to similarity information reflecting collaborative human perception of similarity which is provided by `last.fm`. Using the `last.fm` API function *Artist.getSimilar*[9], we retrieve a list of most similar artists for each artist in the collection. Considering this list as the ground truth, we simulate a retrieval task, interpreting the seed artist as query to the algorithm under evaluation and `last.fm`'s similar artists as relevant items. Both `last.fm` and our algorithms yield ranked lists of closest artists to a given seed. As performance measures we can thus compute *precision–recall* curves and F_1 *measure* (i.e., the harmonic mean of precision and recall).

We first remove from our tweet sets artists who are unknown to `last.fm` since we cannot evaluate them without a ground truth. As some artists appear too infrequently in the microblogs for a reliable evaluation, we further exclude artists who occur less than 50 times in the tweet sets, which roughly corresponds to omitting the long tail of artists. We analyze these artists separately in Sect. 5.4. This effectively reduces the number of artists under investigation to 1,677.[10] The average number of similar

[8] http://last.fm.

[9] http://www.last.fm/api/show/artist.getSimilar.

[10] The number of unique artists for whom we could determine more than 50 `Twitter` occurrences is 2,406. Note that 1,677 refers to the artists for whom we have both `last.fm` and enough `Twitter` data. We filtered the remaining 729 artists unknown to `last.fm`, because we cannot evaluate them (they are not present in the ground truth).

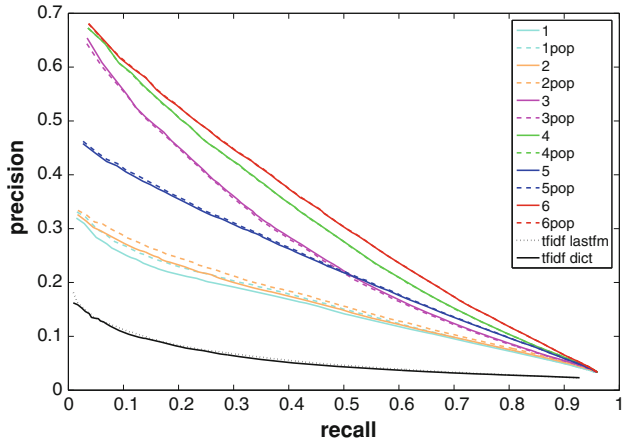

Fig. 3 Precision–recall curves, comparing the 12 proposed co-occurrence similarity functions and classical *tf* · *idf* weighting

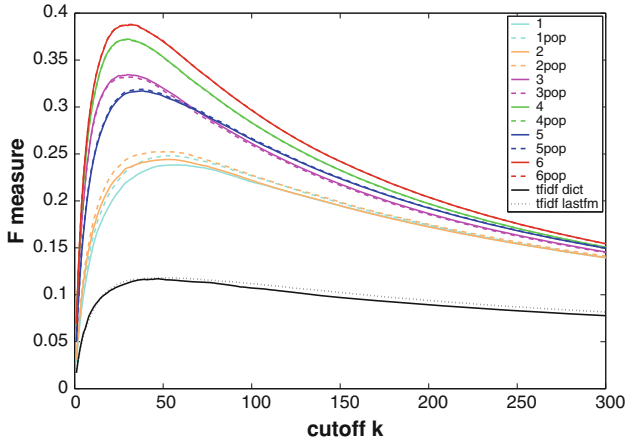

Fig. 4 $F_1@k$ at different cutoffs, comparing the 12 proposed co-occurrence similarity functions and classical *tf* · *idf* weighting

artists returned by `last.fm` is 27.21 for each artist in the entire dataset. The median is 20 since `last.fm` frequently returns only a few similar artists. In our experiments, we include all similar artists provided by `last.fm`, with the exception of those having a weight of zero.

We note that we cannot assure that `last.fm` users do not tweet their listening histories too, so both data sources could actually overlap and therefore invalidate our results to some degree. To minimize this potential bias, tweets including the hashtag `#lastfm`, which is the official way to tweet `last.fm` listening events[11], have not been included when acquiring data (cf. Sect. 3) In fact, the hashtag `#lastfm` only occurs in 11,114 (0.1 %) and 118 (0.02 %) of the tweets in datasets `#nowplaying` and `#itunes`, respectively. This effect can thus be neglected.

5.1.1 Artist retrieval

The results of this similar artist retrieval task are shown in Figs. 3 (precision–recall curves) and 4 (F_1 measure). Two observations can be made immediately. First, the figures show a clear dominance of the proposed co-occurrence approaches over the *tf* · *idf* weighting. This is particularly important because *tf* · *idf* is a standard term weighting approach in Text-IR, which nevertheless seems not suited well to model microblogs. This finding is consistent with [21], which also argues that classical *tf* · *idf* might not be the best choice to model short texts. The second observation is that employing the popularity correction, as proposed in [40], does not improve results. More importantly, it actually hurts performance in some cases. Although formulations *1*, *2*, and *5* benefit from popularity correction, for variants *3*, *4*, and *6*, adding the popularity term does not influence performance or even decreases performance slightly.

An explanation for this effect in formulation *1* is that it does not include the number of occurrences of the second artist *j* (cf. Eq. 1). For variant *2*, the denominator is the minimum of each artist's occurrence. Given the skewed distribution of artist play counts (cf. Figs. 1, 8), the likelihood of picking an artist *i* with smaller popularity, assuming that *j* is a popular one, is much higher than the other way round. This frequently reduces variant *2* to variant *1*, which can be seen particularly well from Fig. 4 (very similar F_1 measures). Comparing formulations *5* and *6*, we see that decreasing the importance of both single occurrence values (by taking the square root in the denominator) considerably increases performance. Another aspect that possibly contributes to the lack of effect of the popularity correction factor is that among similarly popular artists, this factor does not matter anyway. In contrast, if two artists strongly vary in popularity, their similarity score is reduced. But this influences retrieval tasks in different ways depending on whether the query artist is popular or not. In the case where the query consists of a highly popular artist, the correction factor decreases even more the likelihood that unpopular artists are retrieved. On the other hand, when using as query an artist with low popularity, the popularity factor reduces the likelihood of popular artists to be retrieved, to account for the generally higher chance of popular ones to be co-listened to. Therefore, the popularity correction factor only affects the retrieved similar artists for unpopular query artists, if there are both popular and unpopular ones in the list of co-occurring artists.

5.1.2 Rank proximity

For a music retrieval or recommendation system, the order in which results are returned is also important. Even if the

[11] These listening events are named "scrobbles".

actual most similar artists are retrieved, the user will likely not be very content with the retrieval system in case they are always found at the very end of the result set. On the other hand, small differences between the actual rank and the predicted rank do not severely harm the quality of the suggested artist list. To assess this aspect, we compute a rank proximity measure, predicting the k most similar artists (l is the total number of artists in the ground truth), and defining precision and recall equivalents using a weighted rank proximity as follows:

$$prec@k = \frac{1}{k} \cdot \sum_{i=1}^{k} \left(1 - \frac{|i - r_{gt}(a_i)|}{max(i, r_{gt}(a_i))}\right)$$

$$rec@k = \frac{1}{l} \cdot \sum_{i=1}^{k} \left(1 - \frac{|i - r_{gt}(a_i)|}{max(i, r_{gt}(a_i))}\right)$$

where i and $r_{gt}(a_i)$ are the ranks at which artist a_i appears in the predicted ranking and the ground truth, respectively. In contrast to the standard definitions of precision and recall, each true positive in our rank-proximity formulation does not necessarily contribute $1/l$ to recall and $1/k$ to precision, even if there is an overlap, but only if the rank is correct as well. Dividing the absolute rank difference by the maximum rank in ground truth and prediction, we put a stronger penalty on swaps between top-ranked items, which is consistent with user requirements. If a predicted artist is not found in the ground truth, the rank difference, and in turn the penalty, are set to a maximum, i.e., $\frac{|i-r_{gt}(a_i)|}{max(i,r_{gt}(a_i))} = 1$.

Figure 5 reveals that this task is much harder than the standard retrieval task, looking at the range of precision and recall scores achieved. Apart from that, the results are largely comparable to those reported in the previous section. Variants *4*, *4pop*, *5pop*, *6*, and *6pop* tend to outperform variants *1[pop]*, *2[pop]*, *3[pop]* and *5* also in this set of evaluation experiments.

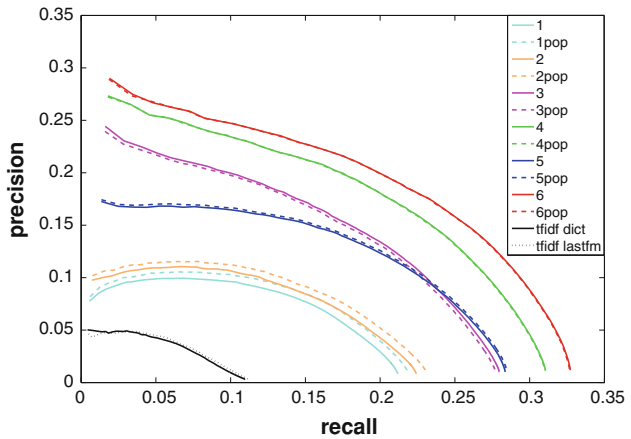

Fig. 5 Precision–recall curves for rank proximity, comparing the 12 proposed co-occurrence similarity functions and classical $tf \cdot idf$ weighting

5.2 Similar artist retrieval evaluated on expert judgments

A potential point of criticism of the previous evaluation approach is the fact that we compare the different scoring functions against another algorithm (by `last.fm`) and not against real human judgments. Although this data source may be regarded as sufficiently valid for our calculations, we additionally compare the similarity measures against real human expert judgments. Unfortunately, it is impossible to get a comparably large amount of human-annotated similarity data.

The best source of human-annotated similarity we could come up with was data from the MIREX[12]. "Audio Music Similarity and Retrieval" (AMS) task. An AMS system is intended to retrieve songs musically similar to a query song. Implicitly, two artists are similar to the extent their songs are similar to each other, so we can use song-similarity data to infer artist-similarity data. Previous work showed evidence that the average similarity of an artist's songs is a very discriminative variable to predict similarity of a new song [37]. Therefore, for any two artists, we compute their similarity as the average similarity between their songs. These similarities between songs are based on actual judgments by human experts, who assess how similar two songs are based on two scales. In the Broad scale, they indicate whether two songs are not similar, somewhat similar, or very similar; while in the Fine scale they provide a similarity score from 0 to 100. We compute the average Fine score over songs and consider the two respective artists similar if this score is greater than 25. This threshold is fixed based on the distribution of Fine scores across Broad scores (see Fig. 1-bottom in [16]).

In total, there are 7,000 songs from 10 major music genres and from 602 unique artists in the MIREX dataset. Our `#nowplaying` and `#itunes` datasets include 327 of these artists. That makes a total of 53,301 possible artist–artist pairs, but we only had judgments to compute similarity in 4,877 of those cases. We used all judgments made during MIREX 2007, 2009, 2010, 2011, and 2012, which account for a grand total of 12,051 judgments among the 327 artists. On average, the similarity between two artists is computed based on three judgments between their songs. For the `#nowplaying` dataset, we could perform an evaluation on 280 artists.

[12] MIREX ("Music Information Retrieval Evaluation eXchange") is the premier annual campaign to evaluate Music IR systems for a variety of tasks. More information, as well as data, can be accessed online at http://music-ir.org/mirex/wiki/MIREX_HOME.

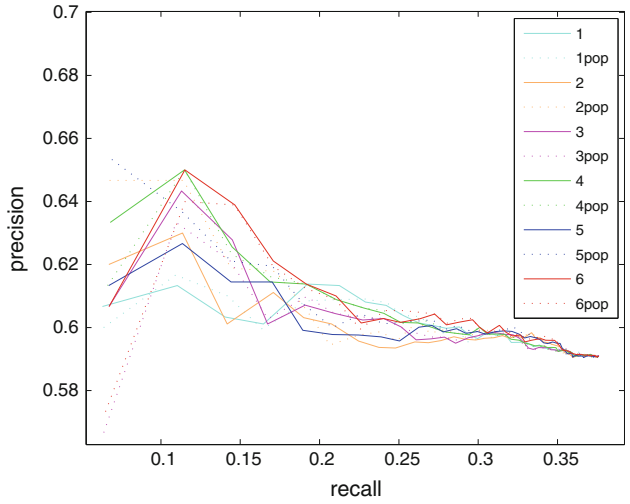

Fig. 6 Precision–recall curves, comparing the 12 proposed co-occurrence similarity functions for the `#nowplaying` dataset using MIREX data as the ground truth

Using the same approach as in Sect. 5.1, we calculated precision and recall for all evaluated scoring functions. As Fig. 6 shows, the same dominance of some functions above others can be observed. Although due to the small number of remaining artists the differences cannot be seen as clearly as in Fig. 3, variant *6* performs best. Due to the smaller number of artists, popularity correction has a stronger noise effect, but again it does not show real improvements.

Summing up, it can be said that although there are few expert judgments available compared to collaborative methods, we show that our findings are consistent in both cases, which supports large-scale experiments using `last.fm` data as the ground truth.

5.3 Time consistency

To investigate whether results are consistent over time, we created a second dataset crawled from December 2011 to March 2012. To base this investigation on a larger corpus, we did not restrict the crawling to tweets with geospatial information ($\approx$ 3 % of the tweets), but used the hashtags `#nowplaying` and `#itunes` as sole filters (resulting in 35 million retrieved tweets).

We employed the same pattern-based approach as presented in Sect. 3 to detect artists. For the original dataset (crawled in 2011), we were able to map 2,945,780 (29.7 %) tweets to specific tracks. On the new dataset, we could assign a song to 6,652,500 (19 %) tweets.

Table 3 shows for both data collections the number of artists remaining for analysis, using different thresholds t_{ao} for minimum artist occurrence in the tweet set. Figure 7 depicts precision–recall curves comparing the performance of all six scoring functions with both 2011 and 2012 datasets. Investigating the figures, we can see similar results for both datasets. The performance of the scoring functions seemingly depends on the threshold t_{ao}, but not heavily on the number of artists, which is quite different for the two datasets (cf. Table 3). Especially, the two dominating scoring functions, *4* and *6*, scale well and show similarly good performance for both collections. Scoring functions *1*, *2*, and *5* perform slightly different for the different datasets.

Table 3 Number of artists in the main dataset gathered in 2011 and a collection gathered in 2012, using different thresholds for minimum artist occurrence t_{ao}

t_{ao}	2011	2012
5	6,465	11,500
50	1,677	3,928
500	257	966

To quantitatively assess the temporal stability of the results, we compute Pearson's correlation coefficient, for each of the 12 similarity functions, between the average $F_1@k$ scores obtained on the 2011 collection and on the 2012 collection. The mean correlation coefficient over all pairs is 0.979. For the best-performing scoring functions *4* and *6*, the mean correlation even exceeds 0.99. We can hence conclude that the relative results are highly stable over time, which is particularly remarkable because of the different number of artists covered by the two datasets (for same t_{ao} values).

5.4 Influence of the long tail

Artists in the "long tail" are those who are listened to infrequently. This is the reason why there is commonly a lack of data for such artists. In turn, they are often neglected when building music retrieval or recommendation systems, as it is particularly challenging to determine suited artists based on a sparse data basis [8].

To illustrate the long tail, we depicted in Fig. 8 the sorted total artist occurrences (as identified by our pattern-based approach) in tweet set `#nowplaying`.[13] Please note that both axes are logarithmically scaled. Since in MIR there are no commonly agreed boundaries on where to split between long tail, mid range, and short head, we quantify different popularity ranges in the following. The top 500 (1.3 %) artists account for 56 % of the listening events ("short head"), while the bottom 20,000 artists (52 %) account for only 2.6 % of the total listening events. If we split the total aggregated number of play counts into three equally sized ranges (each accounting for about 900,000 listening events), one-third of all listening events

[13] The plot looks similar for dataset `#itunes`.

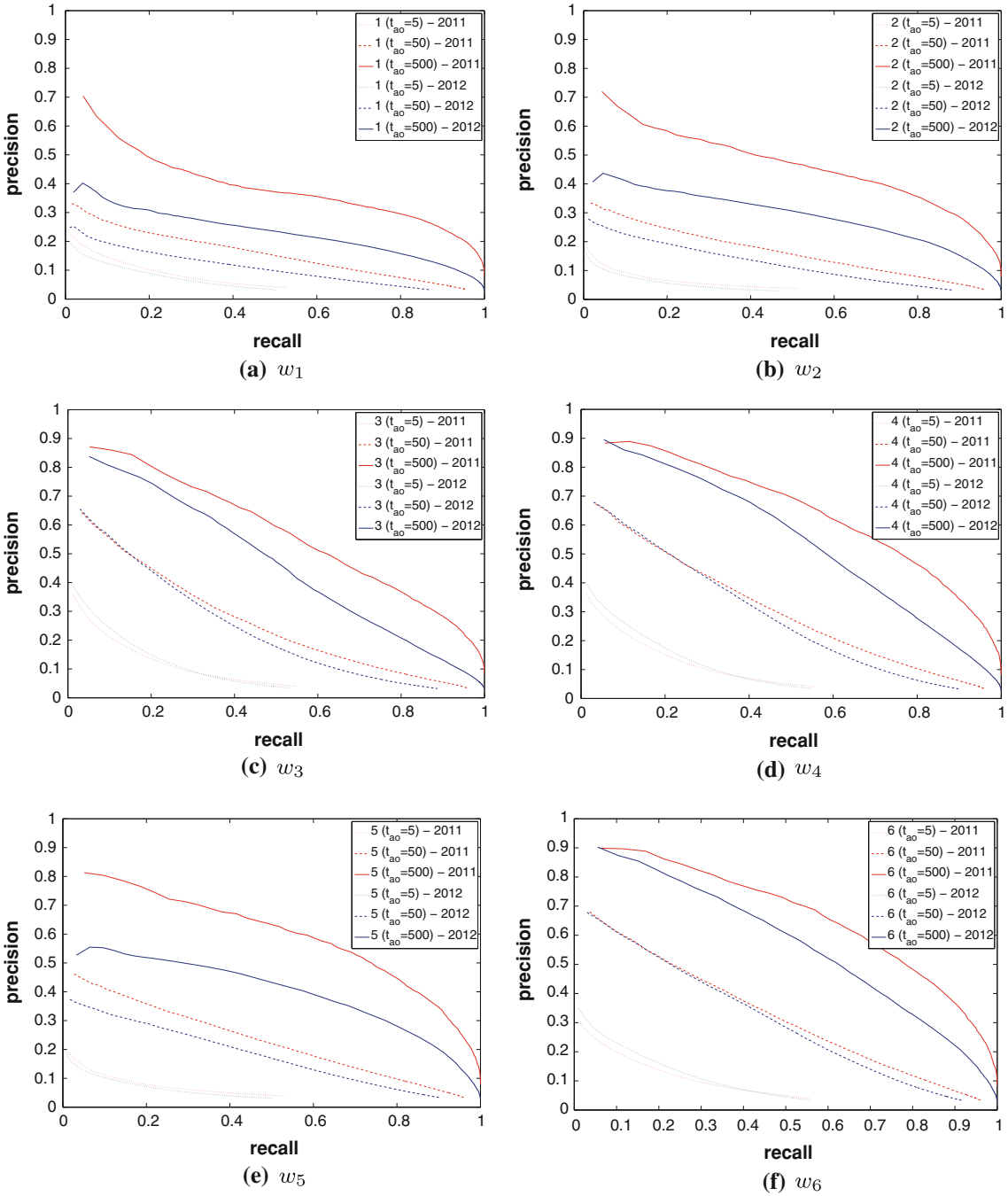

Fig. 7 Precision–recall curves over time (2011 vs. 2012 dataset), comparing the six scoring functions without popularity correction and employing different thresholds for minimum artist occurrence

include artists in range [1, 133], one-third in range [134,926], and one-third in [927,38183]. We finally also indicate in Fig. 8 the positions of the different threshold values for t_{ao} used in the experiments.

To investigate whether the long tail factor has a negative effect on the performance of music retrieval systems based on microblog co-occurrences, we compute the maximum F_1 scores for different levels of the threshold t_{ao}, as these levels correspond to different ranges of artist popularity. Table 4 shows the results for each scoring function, revealing that retrieval accuracy seems indeed highly dependent on artist popularity. Figure 9 shows the results of a more detailed analysis by depicting the average F_1 score (over all artists) dependent on the value of t_{ao}. The figure supports the conjecture from Table 4 that retrieval performance is proportional to the threshold t_{ao}, and hence to artist popularity, and is inversely proportional to the number of artists in the corpus.

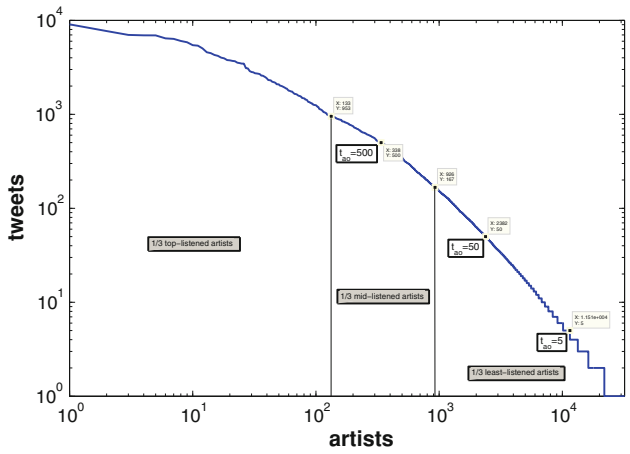

Fig. 8 Number of listening events identified in microblogs for all artists

Table 4 Maximum F_1 scores for different thresholds t_{ao} and scoring functions

	1	2	3	4	5	6
$t_{ao} = 5$	0.122	0.088	0.165	0.171	0.108	0.155
$t_{ao} = 50$	0.238	0.244	0.335	0.372	0.317	0.388
$t_{ao} = 500$	0.447	0.511	0.554	0.618	0.604	0.636

Table 5 Top 20 artists on `Twitter`, using `#nowplaying` and `#itunes`

`#nowplaying`		`#itunes`	
Artist	Tweets	Artist	Tweets
Paramore	9,066	The Beatles	939
Drake	7,697	Daft Punk	683
Katy Perry	6,998	Britney Spears	567
Bruno Mars	6,932	Adele	462
Lady Gaga	6,919	Coldplay	428
Coldplay	6,434	Bruno Mars	416
Eminem	6,352	Katy Perry	374
Rihanna	6,038	The Black Eyes Peas	373
Taylor Swift	5,844	Kanye West	367
Usher	5,445	Lady Gaga	358
Muse	5,383	Avril Lavigne	308
Justin Bieber	5,028	Arcade Fire	299
The Beatles	4,579	Radiohead	266
Michael Jackson	4,476	Kings of Leon	240
Linkin Park	4,285	Duran Duran	238
Oasis	4,190	Michael Jackson	229
Kanye West	4,013	Linkin Park	228
Chris Brown	3,943	Eminem	211
Avril Lavigne	3,780	Muse	209
Radiohead	3,756	The Black Keys	203

Fig. 9 Average F_1 scores for different thresholds t_{ao} and scoring functions

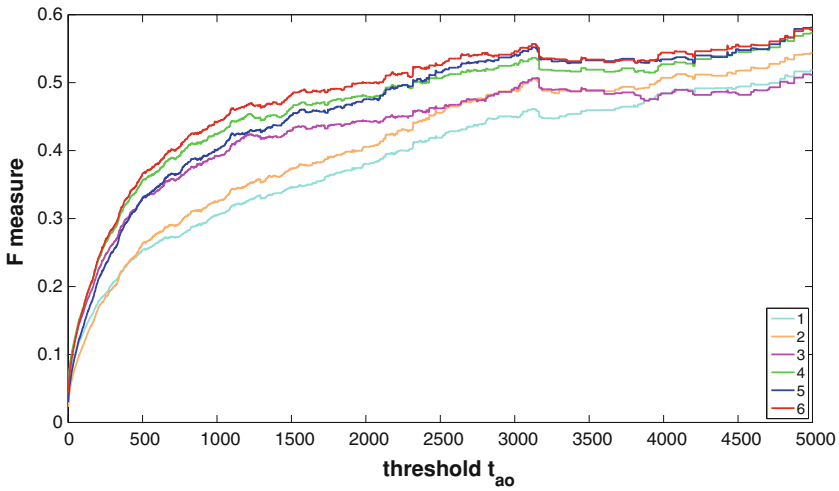

Table 5 lists the top 20 "short head" artists for tweet collections `#nowplaying` and `#itunes`.[14] Most people will agree that the majority of artists are indeed well known. The table also reveals that users of `iTunes` have different music tastes, on average, than users consuming music via other programs or devices.

[14] We cannot give a similar list for the long tail as 16,252 artists in `#nowplaying` (6,021 in `#itunes`) only have a single occurrence.

6 Conclusions and future work

We presented a framework to infer music artist similarity from microblogs and based on co-occurrence information. Evaluating different scoring functions and the use of a correction factor for highly popular artists showed that these co-occurrence-based approaches outperformed traditional $tf \cdot idf$ functions. However, employing a popularity correction factor did not produce a consistent improvement

of results. In fact, it only improved the scoring functions that performed worse anyway, while the best ones were not affected.

Furthermore, we could clearly make out a "long tail" effect shown by considerably higher precision and recall scores for popular artists, regardless of the scoring function. Analyzing the time stability of the results, we gathered a second dataset about 1 year after the first one and demonstrated that results were highly comparable over time.

Future work will aim at combining the context-based methods explored in this paper with audio signal-based approaches to improve accuracy of music retrieval systems. Moreover, we strive to improve artist identification in microblogs. We will further look into aspects other than similarity, for instance, popularity or novelty, which can be derived from microblogs and subsequently used to refine and personalize music retrieval algorithms. In addition, a comparison with approaches that infer similarity from artist co-occurrences on web pages will be performed.

Acknowledgment Spanish Government (HAR2011-27540).

References

1. Armentano, M.G., Godoy, D., Amandi, A.A.: Recommending information sources to information seekers in twitter. In: Proceedings of the IJCAI 2011: International Workshop on Social Web Mining. Barcelona, Spain (2011)
2. Aucouturier, J.J., Pachet, F.: Representing musical genre: a state of the art. J. New Music Res. **32**(1), 83–93 (2003)
3. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. Addison Wesley, Boston (1999)
4. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Lüke, K.H., Schwaiger, R.: InCarMusic: context-aware music recommendations in a car. In: Proceedings of the International Conference on Electronic Commerce and Web Technologies (EC-Web), Toulouse, France (2011)
5. Baumann, S., Hummel, O.: Using cultural metadata for artist recommendation. In: Proceedings of the 3rd International Conference on web delivering of music (WEDELMUSIC 2003). Leeds, UK (2003)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Mach. Learn. Res. **3**, 993–1022 (2003)
7. Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-based music information retrieval: current directions and future challenges. Proc. IEEE **96**, 668–696 (2008)
8. Celma, O.: Music recommendation and discovery—the long tail, long fail, and long play in the digital music space. Springer, Berlin (2010)
9. Celma, O., Cano, P., Herrera, P.: SearchSounds: An Audio Crawler Focused on weblogs. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006). Victoria, Canada (2006)
10. Cimiano, P., Handschuh, S., Staab, S.: Towards the Self-annotating Web. In: Proceedings of the 13th International Conference on World Wide Web (WWW 2004), pp. 462–471. ACM Press, New York, NY, USA (2004)
11. Cimiano, P., Staab, S.: Learning by Googling. ACM SIGKDD Explor. Newsl. **6**(2), 24–33 (2004). doi:10.1145/1046456.1046460
12. Cohen, W.W., Fan, W.: Web-collaborative filtering: recommending music by crawling the Web. WWW9 / Comput. Netw. **33**(1–6), 685–698 (2000)
13. Geleijnse, G., Korst, J.: Web-based artist categorization. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006). Victoria, Canada (2006)
14. Govaerts, S., Corthaut, N., Duval, E.: Using search engine for classification: does It still work? In: Proceedings of the IEEE International Symposium on Multimedia (ISM2009): International Workshop on Advances in Music Information Research (AdMIRe 2009). San Diego, CA, USA (2009)
15. Hu, X., Downie, J.S., West, K., Ehmann, A.: Mining music reviews: promising preliminary results. In: Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005). London, UK (2005)
16. Jones, M.C., Downie, J.S., Ehmann, A.F.: Human similarity judgments: implications for the design of formal rvaluations. In: International Conference on Music Information Retrieval, pp. 539–542 (2007)
17. Knees, P., Pampalk, E., Widmer, G.: Artist Classification with Web-based data. In: Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004), pp. 517–524. Barcelona, Spain (2004)
18. Knees, P., Pohle, T., Schedl, M., Widmer, G.: A Music search engine built upon audio-based and Web-based similarity measures. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007). Amsterdam, The Netherlands (2007)
19. Knees, P., Schedl, M., Pohle, T., Widmer, G.: An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the Web. In: Proceedings of the 14th ACM International Conference on Multimedia (MM 2006). Santa Barbara, CA, USA (2006)
20. McFee, B., Lanckriet, G.: The natural language of playlists. In: Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011). Miami, FL, USA (2011)
21. Metzler, D., Dumais, S., Meek, C.: Similarity measures for short segments of text. In: Proceedings of the 29th European Conference on Information Retrieval (ECIR 2007). Rome, Italy (2007)
22. Naveed, N., Gottron, T., Kunegis, J., Alhadi, A.C.: Searching microblogs: coping with sparsity and document quality. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011), pp. 183–188 (2011)
23. Pampalk, E., Goto, M.: MusicRainbow: A new user interface to discover artists using audio-based similarity and Web-based labeling. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006). Victoria, Canada (2006)
24. Peat H.J., Willett P. (1991) The limitations of TermCo-occurrence data for query expansion in document retrievalsystems. J. Am. Soc. Inform. Sci. Technol. 42:378–383
25. Pohle, T., Knees, P., Schedl, M., Pampalk, E., Widmer, G.: "Reinventing the wheel": a novel approach to music player interfaces. IEEE Trans. Multimedia **9**, 567–575 (2007)
26. Ramzan, N., Zwol, R., Lee, J.S., Clüver, K., Hua, X.S. (eds): Social Media Retrieval. Springer, Berlin (2012)
27. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975). doi:10.1145/361219.361220

28. Schedl, M.: On the use of microblogging posts for similarity estimation and artist labeling. In: Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010). Utrecht, The Netherlands (2010)
29. Schedl, M.: Analyzing the potential of microblogs for spatio-temporal popularity estimation of music artists. In: Proceedings of the IJCAI 2011: International Workshop on Social Web Mining. Barcelona, Spain (2011)
30. Schedl, M.: Music data mining, chap. Web-based and community-based music information extraction. CRC Press/Chapman Hall, Boca Raton (2011)
31. Schedl, M.: #nowplaying Madonna: a large-scale evaluation on estimating similarities between music artists and between movies from microblogs. Inf. Retr. **15**, 183–217 (2012)
32. Schedl, M.: Leveraging microblogs for spatiotemporal music information retrieval. In: Proceedings of the 35th European Conference on Information Retrieval (ECIR 2013). Moscow, Russia (2013)
33. Schedl, M., Hauger, D.: Mining microblogs to infer music artist similarity and cultural listening patterns. In: Proceedings of the 21st International World Wide Web Conference (WWW 2012): 4th International Workshop on Advances in Music Information Research (AdMIRe 2012). Lyon, France (2012)
34. Schedl, M., Knees, P.: Personalization in multimodal music retrieval. In: Proceedings of the 9th Workshop on Adaptive Multimedia Retrieval (AMR 2011). Barcelona, Spain (2011)
35. Schedl, M., Knees, P., Böck, S.: Investigating the similarity space of music artists on the micro-blogosphere. In: Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011). Miami, FL, USA (2011)
36. Schedl, M., Knees, P., Widmer, G.: A Web-based approach to assessing artist similarity using co-occurrences. In: Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing (CBMI 2005). Riga, Latvia (2005)
37. Urbano, J., Schedl, M.: Minimal test collections for low-cost evaluation of audio music similarity and retrieval systems. Int. J. Multimedia Inform. Retr. **2**(1), 59–70 (2013)
38. Vapnik, V.N.: The nature of statistical learning theory. Springer, Berlin (1995)
39. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: Proceedings of the 3th ACM International Conference on Web Search and Data Mining (WSDM 2010). New York, NY, USA (2010)
40. Whitman, B., Lawrence, S.: Inferring descriptions and similarity for music from community metadata. In: Proceedings of the 2002 International Computer Music Conference (ICMC 2002), pp. 591–598. Göteborg, Sweden (2002)
41. Zadel, M., Fujinaga, I.: Web services for music information retrieval. In: Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004). Barcelona, Spain (2004)