Exploring the Music Similarity Space on the Web

MARKUS SCHEDL, TIM POHLE, PETER KNEES, and GERHARD WIDMER,
Johannes Kepler University

14

This article comprehensively addresses the problem of similarity measurement between music artists via text-based features extracted from Web pages. To this end, we present a thorough evaluation of different term-weighting strategies, normalization methods, aggregation functions, and similarity measurement techniques. In large-scale genre classification experiments carried out on real-world artist collections, we analyze several thousand combinations of settings/parameters that influence the similarity calculation process, and investigate in which way they impact the quality of the similarity estimates. Accurate similarity measures for music are vital for many applications, such as automated playlist generation, music recommender systems, music information systems, or intelligent user interfaces to access music collections by means beyond text-based browsing. Therefore, by exhaustively analyzing the potential of text-based features derived from artist-related Web pages, this article constitutes an important contribution to context-based music information research.

Categories and Subject Descriptors: H.4 [**Information Systems**]: Information Systems Applications; H.3 [**Information Systems**]: Information Storage and Retrieval

General Terms: Algorithms, Experimentation, Measurement

Additional Key Words and Phrases: Music information retrieval, Web content mining, term space, evaluation

ACM Reference Format:
Schedl, M., Pohle, T., Knees, P., and Widmer, G. 2011. Exploring the music similarity space on the Web. ACM Trans. Inf. Syst. 29, 3, Article 14 (July 2011), 24 pages.
DOI = 10.1145/1993036.1993038 http://doi.acm.org/10.1145/1993036.1993038

1. INTRODUCTION

Music Information Retrieval (MIR) is a steadily growing field of research. Although early work on how to apply information retrieval (IR) techniques to music dates back to the 1960s [Kassler 1966], MIR's broad emergence as a scientific discipline originates in the late 1990s, when computational power, network bandwidth and storage capabilities reached levels that made feasible signal-based processing and analysis of digital music data. As pointed out in Downie [2003], MIR is a multidisciplinary research endeavor that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world's vast amount of music accessible to all. MIR hence comprises actions, methods, and procedures for recovering stored data to provide information on music [Fingerhut 2004]

This research is supported by the Austrian Science Funds under project numbers L511-N15, P22856-N23, and Z159.
Authors' address: Johannes Kepler University, Department of Computational Perception, Altenberger Straße 69, 4040 Austria; email: markus.schedl@jku.at.

DOI 10.1145/1993036.1993038 http://doi.acm.org/10.1145/1993036.1993038

and is concerned with the extraction, analysis, and usage of information about any kind of music entity (for example, a song or a music artist) on any representation level (for example, audio signal, symbolic MIDI representation of a piece of music, or name of a music artist) [Schedl 2008].

These definitions of MIR already indicate that it is a highly dynamic and multidisciplinary field of research that relates to various other research disciplines. Narrowing the focus to information extraction (IE) and information representation related to music, we can distinguish three broad categories of strategies in terms of the underlying data source, namely *music content*-based, *music context*-based and *user context*-based approaches. Feature vectors describing aspects from one or more of these three categories can be constructed, and similarity measures can be applied to the resulting vectors of two pieces of music or two music artists.[1] Elaborating such musical *similarity measures* that are capable of capturing aspects that relate to real, perceived similarity is one of the main challenges in MIR. At this point the reader may ask why music similarity is such an important concept. First, music similarity measures can help to understand why two music pieces or artists are perceived alike by the listener. In fact, a listener may state that two songs resemble each other, but cannot tell why they are similar. In this case, computational music similarity measures could give an explanation. Second, similarity measures are of particular importance in the music domain, because, unlike in image retrieval, where the viewer can mentally process the main content of an image within 150 msecs [Thorpe et al. 1996], a piece of music requires a much longer processing time by the auditory system. Music similarity measures are hence important to guide the user in efficiently retrieving a desired piece of music. Consequently, they are a key ingredient of various music-related applications. Examples are systems to automatically generate playlists [Aucouturier and Pachet 2002; Pohle et al. 2007c], music recommender systems [Celma and Lamere 2007; Zadel and Fujinaga 2004], music information systems [Schedl 2008], semantic music search engines [Knees et al. 2007], and intelligent user interfaces [Pampalk and Goto 2007; Knees et al. 2007] to access music collections by means more sophisticated than the textual browsing facilities (`artist-album-track` hierarchy) traditionally offered.

Methods to derive content-based features [Casey et al. 2008] extract information from a data source that represents the content of a piece of music. Most frequently, this is some manifestation of a song's audio signal, for instance, an mp3 file. Such content-based methods allow to model certain aspects of music that relate to acoustic properties. They are capable of describing, for example, the timbre ("sound") of a piece of music [Aucouturier and Pachet 2004] or its rhythmical structure [Pampalk et al. 2002; Schedl et al. 2005]. Recent work addresses more specific high-level aspects, such as melodiousness and "percussiveness," that is, the strength of percussive sounds in the signal [Pampalk 2006; Pohle 2009; Seyerlehner et al. 2007].

Music content-based approaches, however, fall short of describing some aspects that are important to the perception and understanding of music, but are not encoded in the audio signal. For example, an artist's geographic and cultural context, the political background, or the meaning of the song lyrics are likely to influence how his or her music is perceived, but cannot be detected from the music content. Therefore, an analysis of the music context [Schedl and Knees 2009] is necessary if we aim at distilling such factors. Among many other data sources, some of which will be presented as part of related work, an obvious source for contextual data is the World Wide Web. First steps towards using Web pages to derive term feature vectors for the purpose of artist similarity calculation were undertaken in Cohen and Fan [2000a], Whitman and Lawrence

[1]For reasons of simplicity, we use the term "artist" in the following to denote individual performers as well as bands performing music.

[2002], and Knees et al. [2004]. In this work the authors usually select a specific variant of the $tf \cdot idf$ term weighting measure [Baeza-Yates and Ribeiro-Neto 1999] and apply it to Web pages 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. They rather resemble common variants that are known to yield good results in IR tasks. Whether these variants are also suited to describe music artists via term profiles and subsequently estimate similarities between them is seldom assessed comprehensively in the literature on text-based music information extraction.

Addressing this lack of investigation, we present the first comprehensive study on Web-based music similarity estimation. Our work is inspired by Zobel and Moffat [1998], where the authors thoroughly evaluate various decisions involved in constructing text-based feature vectors for IR purposes, for instance, term frequency, term weights (idf), and normalization approaches. They analyze the influence of these decisions on retrieval behavior. Similarly, we present a large-scale study on the influence on similarity estimation of a multitude of decisions, using real-world data collections. To this end, we analyze several thousand different combinations of the following single aspects:

—term frequency;
—inverse document frequency;
—virtual document modeling;
—normalization with respect to page length;
—similarity function.

The *term frequency* $r_{d,t}$ of a term t in a document d estimates the importance t has for document (related to artist) d. The *inverse document frequency* w_t estimates the overall importance of term t in the whole corpus and is commonly used to weight the $r_{d,t}$ factor, that is, downweight terms that are important for many documents and hence less discriminative for d. *Virtual document modeling* relates to the way individual documents retrieved for the same artist are aggregated. We further assess the impact of *normalization* with respect to length of individual Web pages. Different *similarity functions* S_{d_1,d_2} estimate the proximity between the term vectors of two documents or artists d_1 and d_2.

For reasons of completeness, let us state that the third category of MIR-related data sources, the user context, is not directly related to properties of music pieces or artists. It rather comprises external factors that influence how a listener perceives music. Examples for such aspects are the situation in which the listener consumes music (active vs. passive listening, romantic dinner, relaxed evening after a stressful day, preparing to go out on a Saturday night, playing music him/herself in a band), the listener's mood, his or her location, the used listening device (PC, stereo, cell phone, mobile music player), and the listener's social context (friends, peer groups, neighbors, listener's role in the context). In Göker and Myrhaug [2002] a general categorization of user context aspects is presented. However, user context aspects will not be discussed in detail in this contribution.

The remainder of this article is organized as follows. Section 2 outlines the context of this work by conducting a literature review on music context-based similarity estimation. Section 3 then discusses common approaches to extracting music-related information from the Web and details the specific approach we employed. An analysis and discussion of different decisions in the artist description, term weighting, and similarity measurement process can be found in Section 4. Finally, conclusions are drawn in Section 5.

2. BACKGROUND AND RELATED LITERATURE

Estimating similarities between music artists can be performed based on three categories of data sources: music content, music context, and user context. Since we investigate the use of Web pages to derive similarities in this article, we will review related work on Web-based music information extraction methods. Since Web pages reflect human knowledge and opinions, such methods hence fall into the category of music context-based approaches.

2.1. Explicit Similarity Data Collection

The most straightforward way to collect information about artist similarity, or related information such as genre, is by letting people explicitly deliver it. For example, Berenzweig et al. [2003] present a Web-based user survey asking people about their similarity judgments in a set of 400 artists.

Another source of musical knowledge is expert opinions. The music information system allmusic.com, for example, provides for each artist a list of similar artists and a list of genres the artist is assigned to. In a number of publications, such expert opinions have been used as ground truth to evaluate automated approaches [Berenzweig et al. 2003; Ellis 2002].

In recent years, *tagging* has become more popular. For example, the online music platform last.fm lets users assign tags to pieces of music, or music artists. This tag data is made available via an API. Another approach is to collect tags in the form of a game [Law et al. 2007; Mandel and Ellis 2007; Turnbull et al. 2007; Turnbull et al. 2008]. The basic principle of the tagging game [Ahn and Dabbish 2004] is to present the same item (which is a song in this case) to two different players, asking them to provide tags. Points are rewarded when both users provide matching tags. Tags that are proposed multiple times are taken as valid annotations for the item.

2.2. User Collections and Playlists

While explicitly asking people to provide similarity information is usually a source of high-quality data, it is also a very time-consuming task. A less time-consuming alternative to obtain certain types of information about music is to analyze user data, such as which music users have in their music collection, and how often they listen to which artists or songs. For example, Whitman and Lawrence [2002] calculate artist similarity based on cooccurring artists shared by users of the Peer-to-Peer (P2P) network OpenNap.[2] In a more recent work Shavitt and Weinsberg [2009] derive similarity information at the artist level and at the song level from the Gnutella P2P file sharing network. Shavitt and Weinsberg collected metadata of shared files from more than 1.2 million Gnutella users in November 2007, restricting their search to music files (.mp3 and .wav). The crawl yielded a data set of 530,000 songs. They used the data for song clustering and artist recommendation.

Alternatively, music playlists can be analyzed for track or artist cooccurrence patterns [Logan et al. 2003; Stenzel and Kamps 2005]. Playlists created by human users can be obtained, for example, from artofthemix.org or mixtape.me. Exploiting playlists to derive artist similarity information is performed in Baccigalupo et al. [2008], where the authors analyzed cooccurrences of artists in playlists shared by members of a Web community. The authors looked at more than 1 million playlists made publicly available by MusicStrands,[3] a Web service (no longer in operation) that allowed users to share playlists. The authors extracted the 4,000 most popular artists from the full playlist

[2]http://opennap.sourceforge.net.
[3]http://music.strands.com.

set, measuring the popularity as the number of playlists in which each artist occurs. They further take into account that two artists that consecutively occur in a playlist are probably more similar than two artists that occur farther away in a playlist. The authors use this data to define fuzzy genre membership of artists.

2.3. Song Lyrics

The lyrics of a song represent an important aspect of the semantics of music since they are typically closely tied to the artist or the performer by revealing, for example, cultural background, political orientation, or style of music (use of a specific vocabulary in certain music styles).

Logan et al. [2004] use lyrics of songs by 399 artists to determine artist similarity. To this end, in a first step, Probabilistic Latent Semantic Analysis [Hofmann 1999] is applied to a collection of over 40,000 song lyrics to extract N topics typical to lyrics. In a second step, all lyrics by an artist are processed using each of the extracted topic models to create N-dimensional vectors of which each dimension gives the probability of the artist's tracks to belong to the corresponding topic. Artist vectors are then compared by calculating the L_1 distance (also known as Manhattan distance). Evaluation is performed against human similarity judgments, that is, the "survey" data for the uspop2002 set [Berenzweig et al. 2003]. Logan et al.'s approach does not reach performance levels similar to those obtained via acoustic features (irrespective of the chosen N, the usage of stemming, or the filtering of lyrics-specific stopwords). However, as lyrics-based and audio-based approaches make different errors, a combination of both is suggested.

Mahedero et al. [2005] demonstrate the usefulness of lyrics for similarity measurement, among other tasks. A standard $tf \cdot idf$ measure with cosine distance is proposed as initial step. Using this information, a song's representation is obtained by concatenating distances to all songs in the collection into a new vector. These representations are then compared using an unspecified algorithm. Exploratory experiments indicate some potential for cover version identification and plagiarism detection.

The goal of Laurier et al. [2008] is classification of songs into four mood categories by means of lyrics and content analysis. For lyrics, the $tf \cdot idf$ measure with cosine distance is incorporated. Optionally, also Latent Semantic Analysis [Deerwester et al. 1990] is applied to the $tf \cdot idf$ vectors (achieving best results when projecting vectors down to 30 dimensions). In both cases, a 10-fold cross validation with k-nearest neighbor (k-NN) classification yielded accuracies slightly above 60%. Audio-based features performed better compared to lyrics features, however, a combination of both yielded best results.

Hu et al. [2009] experiment with $tf \cdot idf$, tf, and Boolean vectors and investigate the impact of stemming, part-of-speech tagging, and function words for soft-categorization into 18 mood clusters. Best results are achieved with $tf \cdot idf$ weights on stemmed terms. An interesting result is that in this scenario, lyrics-based features alone can outperform audio-based features.

3. WEB PAGE ANALYSIS

This section reviews a number of ways to obtain data relevant for music retrieval from the Web. Furthermore, our specific Web-based approach to automatically deriving information about similarity of music artists is presented. By querying a search engine, a number of Web pages is collected for each artist, and the subsequent use of text mining techniques allows for computing a similarity score between two artists.

When it comes to deriving artist-related information from the Web, usually all Web pages returned for a particular artist are regarded as one large, virtual document describing the artist under consideration. This aggregation seems reasonable since, in Web-based MIR, the usual entity of interest is the music artist, not a single Web page.

Furthermore, it is easier to cope with very small, or even empty, pages if they are part of a larger virtual document.

The process of obtaining music-related metadata from the Web by using text information retrieval techniques can be divided into three stages: data acquisition, data analysis and usage, which are discussed in the following.

3.1. Data Acquisition

The first step towards building a Web-based music similarity measure consists of identifying Web pages related to the music domain, for example, fan pages, biographies, album reviews, track lists, or sale offers for albums or songs. This Webpage selection can be carried out either by using a *focused crawler* [Chakrabarti et al. 1999] or by relying on Web search engines. Using a specialized focused crawler has the potential of yielding better pages as it intends to effectively confine the crawl to the music domain. However, since it involves various complex components (e.g., link analyzer and classifier), computational performance is generally limited. Issuing queries to a Web search engine to obtain related pages, in contrast, is fast and easy. On the other hand, the number of allowed automatically sent queries is usually limited and the ranking algorithm applied by the search engine is in most cases a well-kept secret.

Automatically querying a Web search engine to determine pages related to a specific topic is a common and intuitive task, which is therefore frequently performed in IE research. Examples in the music domain can be found in Whitman and Lawrence [2002] and Geleijnse and Korst [2006], whereas Cimiano et al. [2004], Cimiano and Staab [2004], and Knees et al. [2007] apply this technique in a more general context. Although this approach seems to be straightforward, it is prone to a major type of error: When searching for artist names that equal common speech words, usually a lot of irrelevant pages are returned.[4] Hence, the main challenge is to restrict the search results to pages related to the desired artist. This problem is commonly addressed by enhancing the search query for the artist name with additional keywords. In the context of music information research, Whitman and Lawrence [2002] proposed to confine the search by the keywords "music" and "review" in order to direct it towards album reviews. The resulting query scheme has successfully been applied in genre classification tasks [Knees et al. 2004]. Later research has shown that other keywords seem to yield more accurate results, depending on the task. For example, when aiming at determining band members, the query schemes "*artist*" `music` and "*artist*" `music members` were more successful [Schedl and Widmer 2007]. To gather general, music-related Web pages, the scheme "*artist*" `music` usually represents a good trade-off between coverage and false positives. Hence, we used it for the article at hand. It has to be borne in mind, however, that these settings are not suited for multilingual pages and artists for which no English content is available on the Web. Varying the language of the additional keywords (e.g., `music`, `Musik`, `musique`, `musica`) may resolve this issue, but at the price of considerably increasing the number of queries issued to the search engine. For almost all artists in our test collections, the number of available Web pages is well above the number of actually retrieved ones. Restricting the search to English keywords therefore does not impose any limitations concerning the quantity of artist-related pages analyzed. However, one should be aware that, in general, restricting the search space to English pages might yield undiscovered pages that are nevertheless relevant to the artist.

We first query Google's search engine to retrieve up to the top 100 URLs for each artist in the collection. We then fetch the Web content available at these URLs using an optimized fetcher featuring load balancing, which we implemented in Java.

[4] In the music domain typical artists that cause such problems are *Bush*, *Prince*, *Kiss*, and *Porn*.

Subsequently, we create a *full inverted index*, also known as *world-level index* [Zobel and Moffat 2006], using a modified version of the open-source indexer Lucene Java.[5]

3.2. Text Analysis and Processing

From the kind of data acquired in the previous step, Whitman and Lawrence [2002] extract *unigrams* (single words occurring in the texts), *bigrams* (pairs of words following each other in the texts), words that are likely to be *adjectives* (by applying a part-of-speech (POS) tagger), and noun phrases. Each of these forms a possible basis for a vector space, where each term (e.g., bigram) is one dimension. In Pampalk et al. [2005], as an alternative to generating the space out of the retrieved documents, a predefined dictionary of words is used that are meaningful in the music domain. To cope with different forms of the same word, a stemming algorithm can be used [Celma et al. 2006; Schnitzer et al. 2007] at the expense of potentially introducing ambiguities. In many cases, words that are very frequent (such as *the, I*) and thus are assumed to not carry a meaning in the particular domain are removed by using *stopword lists*.

The actual value $w_{d,t}$ assigned to an artist d in each dimension of the term space is computed from the frequency with which the term t occurs in documents related to this artist (*term frequency*, $r_{d,t}$), and typically is normalized by the count of the number of documents in which the term occurs (*document frequency*, w_t). The resulting vector is generally referred to as $tf \cdot idf$ vector. The basic intention of the tf factor is to assign higher weights to terms that occur more frequently on pages retrieved for artist d, whereas the inverse document frequency idf factor downweights terms that often occur in the whole corpus for different artists and therefore are not specific to artist d. Most formulations of idf apply the logarithm to the raw document frequency values to particularly suppress terms with very high df values (cf. Table III).

The preceding procedure is used to create a $tf \cdot idf$ vector space from the retrieved documents. However, there exist scenarios where other representations are used. For example, for the task of artist recommendation, in Cohen and Fan [2000] lists of artists are extracted from Web pages to eventually construct pseudo-users for a collaborative filtering approach. In Pachet et al. [2001], texts are analyzed for the occurrence of track and artist names to facilitate cooccurrence and correlation analysis for similarity computation. Schedl et al. [2007] combine named entity detection and a rule-based IE approach to derive band memberships. Approaches to predict the geographic origin of an artist are presented in Govaerts and Duval [2009] and Schedl et al. [2010].

An alternative term weighting scheme is the *BM25* function that is used in the Okapi framework for text-based probabilistic retrieval [Robertson et al. 1995; Robertson et al. 1999]. This model assumes a priori knowledge on topics from which different queries are derived. Moreover, based on information about which documents are relevant for a specific topic and which are not, the term weighting function can be tuned to the corpus under consideration. Since *BM25* is a well-established term-ranking method, we included it in the experiments. However, it has to be noted that in our case, we cannot assume any a priori classification, neither on the level of Web pages, nor on the artist level. On the Webpage level, manually classifying hundreds of thousands of Web pages would be too labor-intensive. On the artist level, we could obviously group the artists (or more precisely, the retrieved Web pages of the artists) according to a genre taxonomy and optimize *BM25* correspondingly. However, we believe that this is not justifiable for two reasons: First, for arbitrary music collections, we cannot assume to have genre information given. Second, using genre information would obviously bias the evaluation results for the genre classification experiments as the other term

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

Table I. Denominations for Terms Commonly Used in Text Information Retrieval

$\mathcal{D}$	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
$\mathcal{T}_d$	the set of distinct terms in document d
f_d^m	the largest $f_{d,t}$ of all terms t in d
f^m	the largest f_t in the collection
$r_{d,t}$	term frequency; see Table II
w_t	inverse document frequency; see Table III
W_d	document length of d

weighting measures do not incorporate such a priori knowledge. Thus, *BM25* would be unjustifiably favored.

For our experiments, we therefore used a simpler *BM25* formulation as the one proposed in Robertson et al. [1999], cf. Section 4.1.4.

3.3. Usage

In a number of cases, data usage is tightly coupled with the previous steps (i.e., data retrieval and processing are chosen and designed with a particular application in mind). However, some of the data representations can be used for a variety of applications. Most notably, if a similarity function can be built on the extracted data, potential data usages include clustering, classification, and recommendation. Besides genre classification [Knees et al. 2004], it has been proposed to classify record reviews into classes of "like" and "dislike" [Hu et al. 2005], which eventually could be used to create recommendation systems with improved recommendation performance, for instance, by using only those record reviews that are known to be in line with the user's taste. Another application scenario is a user interface where the user can browse an artist collection via topics automatically derived from $tf \cdot idf$ vectors [Pohle et al. 2007a, 2007b].

In our large-scale analysis of Web-based music artist similarity measures, we derive and evaluate different variants of vector space representations as described in Section 4.

3.4. Similarity Estimation Approaches in Previous Work

A look into the literature reveals that there exist different ways to transform Web pages to a vector of term weights for artists. For example, differences lie in the way basic concepts of text information retrieval, most notably the concept of a *document*, are transferred to music artists who are represented by a number of Web pages. In the following, we use the denominations listed in Table I to refer to various terms of this domain.

Whitman and Lawrence [2002] and Whitman [2005] treat each artist as one document for calculating the document frequency (f_t), while term frequency ($f_{d,t}$) is the percentage of Web pages containing the term. Both f_t and $f_{d,t}$ are normalized, being considered a probability distribution (f_t are normalized after summing up over all artists, while $f_{d,t}$ are normalized for each artist), then $tf \cdot idf$ is computed and normalized for each artist individually in the range 0..1. Optionally, very frequent and very infrequent terms are downweighted by a Gaussian function. The similarity of two artist vectors is calculated by summing up the weights of terms occurring for both artists.

In Baumann and Hummel [2003] and Knees et al. [2004], $f_{d,t}$ is the number of occurrences of term t on the Web pages related to an artist d, and the document

frequency f_t is the number of Web pages the term occurs on (not the number of artists for which the term occurs).

Baumann and Hummel [2003] and Knees et al. [2004] differ in the way N is defined and the $tf \cdot idf$ vector is calculated, while both use the cosine similarity measure to compare artist vectors. Baumann and Hummel [2003] define N as the size of "the entire artist collection", and $tf \cdot idf$ is computed as

$$w_{d,t} = f_{d,t} \cdot \log\left(\frac{N}{f_t}\right). \tag{1}$$

In Knees et al. [2004], N is the total number of pages that were retrieved. For $tf \cdot idf$ computation the following variant is used:

$$w_{d,t} = \begin{cases} (1 + \log_2 f_{d,t}) \log_2 \frac{N}{f_t} & \text{if } f_{d,t} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

As motivated by these examples, there is no standard way to calculate $tf \cdot idf$ vectors from retrieved Web pages, and it is unclear which way to calculate it is preferable. In the next section, a number of variants to obtain $tf \cdot idf$ vectors (and how to compare them) is evaluated to gain some insight into this question.

4. EVALUATING VARIANTS OF TERM WEIGHTING, NORMALIZATION, AND DISTANCE MEASURES

As outlined above, it is a common technique to obtain descriptions of artists by analyzing the text of Web pages returned by a search engine queried with the artist name (and additional query terms to narrow the search to pages more relevant for the domain of music). This "search engine" approach has several advantages. First, the obtained data can be used in different ways (similarity computation [Knees et al. 2004], tagging artists [Schedl and Pohle 2010], categorizing artists [Geleijnse and Korst 2006], or deriving specific information [Schedl et al. 2007]). Second, this approach does not crucially depend on the availability of a specific online platform providing the particular type of data sought. Also current trends in music (e.g., emerging genres) are likely to be reflected in the returned pages quickly. Furthermore, future advances in indexing and search engine technology (finding more relevant pages related to an artist) can be expected to enhance the results.

In this section, we present the evaluation experiments conducted to assess different algorithm variants for calculating artist similarity based on term feature vectors. To assess the quality of the results, we perform genre classification experiments. Even though musical genre is an ill-defined concept and genre taxonomies tend to be highly inconsistent [Pachet and Cazaly 2000], we unfortunately do not have access to reliable and comprehensive similarity data, against which we could perform comparison. We therefore opted for a genre classification task that serves as proxy for artist similarity. We used a k-NN classifier (leave-one-out), and we investigated classification accuracy for different values of k. The assumption underlying the genre classification setting is that similar artists are assigned to the same genre. In leave-one-out classification, the training set consists of all artists except the one to be classified. For each seed artist a, it is tested whether the k closest neighbors' genre labels match a's genre label (where closeness is measured by the similarity algorithm under evaluation). The majority of a's closest neighbors' genre labels is used to classify a. Classification accuracy is computed as arithmetic mean when taking each artist in the collection as seed once.

4.1. Experimental Setup

For our investigation, we opt for an approach comparable to Zobel and Moffat [1998]. A large number of decisions involved in creating artist feature vectors (such as the choice of term frequency $r_{d,t}$ and inverse document frequency w_t), as well as ways to calculate similarity between such feature vectors are evaluated. Most ways to compute these parts originate from previous work in text information retrieval.

4.1.1. Document Modeling/Aggregating Documents. The most central step is the modeling of fundamental text information retrieval concepts such as documents and term frequencies. Once this step is accomplished, known methods to calculate tf (and idf) can be evaluated. In common IR tasks, each document is considered a separate entity. In contrast, in our task each artist is an entity which is represented by a number of documents (i.e., Web pages). There are several ways how to deal with this situation. We evaluate five of them.

(1) *Sum.* All term frequencies appearing in the Web pages associated with the artist are summed up. This corresponds to a simple concatenation of all Web pages related to the artist to one large document.
(2) *Mean.* The term frequency of a term is calculated by taking the arithmetic mean over all pages retrieved for the artist. This is similar to approach 1, but differs in that it is independent of the number of Web pages actually retrieved. Also the range of values is different, which has an impact on some TF calculation approaches.
(3) *Max.* Take the maximum of each term frequency over all retrieved Web pages for the artist.
(4) *NumPagesRel.* Following Whitman [2005], the number of Web pages (retrieved for the artist) that contain the term is used as term frequency. This number is divided by the total number of pages retrieved for the artist.
(5) *NumPagesAbs.* As approach 4, but with the absolute page count, which has an impact on some TF calculation approaches.

We refer to the representation that results from aggregating a number of Web pages retrieved for an artist as *virtual document*.

4.1.2. Page Length Normalization. Based on the idea that Web pages with many terms (i.e., long Web pages) could dominate shorter but nonetheless relevant pages, additionally a normalization step is performed before these aggregation functions are calculated. To minimize interference with the TF calculation approaches (which may depend on the magnitude of the values), the number of terms in each page is normalized to the *page length* (as measured by the sum over the page's raw term frequency count vector). This optional normalization step is done before calculating the TFs, because it intends to simulate pages of same length.

It should be noted that there is another interesting method to combine the Web pages of one artist. It would be possible to calculate the $tf \cdot idf$ value for each Web page separately (i.e, in the initial setup, each Webpage corresponds to one document), and then combine all pages belonging to one artist by a simple aggregation function such as minimum, mean, median or maximum (which may yield different results than mean, subject to the similarity function used). In this case, these functions are calculated *after* having calculated the $tf \cdot idf$ values. We refrain from using this method because the notion our method is based on is to level out page length, a page being either defined as a single Webpage or a virtual artist document (cf. next section). In contrast, that alternative way to combine pages could rather be seen as an attempt to level out different relevances of the retrieved pages. Differing Webpage relevance is not

Table II. Evaluated Variants to Calculate the Term Frequency $r_{d,t}$

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 Zobel and Moffat [1998] 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 Zobel and Moffat [1998] and Robertson et al. [1995]. For W we use the vector space formulation, that is, 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 Robertson et al. [1999].	$r_{d,t} = \frac{(k_1+1) \cdot f_{d,t}}{f_{d,t} + k_1 \cdot \left[(1-b) + b \cdot \frac{W_d}{av_{d \in D}(W_d)} \right]}$ $k_1 = 1.2, b = 0.75$

considered in our evaluations, as the retrieval of relevant pages is delegated to the search engine.

4.1.3. Modeling Document Frequency. In the experiments, we opted to model document frequency f_t in two ways. The first way is to regard each virtual artist document as an atomic entity (i.e., N is the number of artists, and f_t is based on the "virtual documents", *vd*). The second way is to take the number of Web pages as the number N of documents and perform the calculation of f_t on individual Web pages (*wp*).

4.1.4. Calculating and Combining tf and idf Weights. In our experiments, nine different methods for calculating the term frequency $r_{d,t}$ are evaluated, as given in Table II. Correspondingly, Table III gives the evaluated methods to calculate the inverse document frequency w_t. Table IV lists the evaluated similarity functions. Disregarding redundant settings,[6] a total of 9,248 different combinations can be defined (by varying the page aggregation function, page length normalization, TF approach, way to model document frequency, IDF approach, and similarity measure). It should be kept in mind that it is likely that the considered functions interfere with the (generally unknown) ranking algorithm used by the search engine, and probably also with the query terms [Knees et al. 2008].

4.1.5. Algorithm Notation. One overall artist similarity algorithm is created by choosing from the options discussed above. In the remainder of this article, we denote such an algorithm in the following way:

```
<PageAggregationFunction>.<PageLengthNormalization>.<TF-Approach>.
    <IDF-Document-Type>.<IDF-Approach>.<SimilarityMeasure>
```

An example of an algorithm in this notation is *Sum.NoPlNorm.TF_A.VirtualDoc.IDF_B2.CosSim*. In cases where a particular choice of variant is clear from the

[6]Note that in some cases, distinct combinations yield the same $tf \cdot idf$ vectors. For example, the value of TF_A is not affected by normalization of pages.

Table III. Evaluated Variants to Calculate the Inverse Document Frequency w_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 [Robertson et al. 1999; Pérez-Iglesias et al. 2009]	$w_t = \log\frac{N - f_t + 0.5}{f_t + 0.5}$

Table IV. Evaluated Similarity Functions S_{d_1,d_2}

Abbr.	Description	Formulation
INNER	Inner product	$S_{d_1,d_2} = \sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} \cdot w_{d_2,t}\right)$
COSSIM	Cosine Measure	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} \cdot w_{d_2,t}\right)}{W_{d_1} \cdot W_{d_2}}$
INNER_ALT	Alternative Inner Product	$S_{d_1,d_2} = \sum_{t \in \mathcal{T}_{d_1,d_2}} \frac{w_{d_2,t}}{W_d}$
DICE	Dice Formulation	$S_{d_1,d_2} = \frac{2\sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} \cdot w_{d_2,t}\right)}{W_{d_1}^2 + W_{d_2}^2}$
JACC	Jaccard Formulation	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} \cdot w_{d_2,t}\right)}{W_{d_1}^2 + W_{d_2}^2 - \sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} \cdot w_{d_2,t}\right)}$
OVER	Overlap Formulation	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} \cdot w_{d_2,t}\right)}{\min(W_{d_1}^2, W_{d_2}^2)}$
EUCL	Euclidean Similarity	$D_{d_1,d_2} = \sqrt{\sum_{t \in \mathcal{T}_{d_1,d_2}}\left(w_{d_1,t} - w_{d_2,t}\right)^2}$ $S_{d_1,d_2} = \left(\max_{d_1',d_2'}(D_{d_1',d_2'})\right) - D_{d_1,d_2}$
JEFF	Jeffrey Divergence-based Similarity	$S_{d_1,d_2} = \left(\max_{d_1',d_2'}(D_{d_1',d_2'})\right) - D_{d_1,d_2}$ $D(F,G) = \sum_i\left(f_i \log\frac{f_i}{m_i} + g_i \log\frac{g_i}{m_i}\right)$ $m_i = \frac{f_i + g_i}{2}$

context (e.g., when only considering algorithms without page length normalization), the respective part is left out in the notation for brevity.

4.1.6. Term Dictionary. In the literature, there exist a variety of ways to define the terms associated with each dimension of the vector space. To not further complicate the experiments, we opt for using a manually defined dictionary containing 1,379 music-related terms. Assuming that the way of choosing the dictionary avoids common stopwords and terms that appear very infrequently, no downweighting of very frequent and very rare terms is performed. The dictionary comprises terms related to the music domain, such as genre and style descriptors, instruments, epochs, regions, and moods. It was compiled by extracting and merging lists from various Web sources, such as Yahoo! Directory,[7] Wikipedia,[8] and allmusic.com.[9] The list is available for download.[10]

4.1.7. Models Closest to Previous Work. To give a rough orientation how the evaluated techniques are associated with previously used combinations, the closest models to [Whitman and Lawrence 2002; Baumann and Hummel 2003, Knees et al. 2004, Whitman 2005] are given here:

The model closest to Baumann and Hummel [2003] is *Sum.TF_B.NoPlNorm.IDF_B2.CosSim*,[11] and the closest to Knees et al. [2004] is *Sum.TF_C.NoPlNorm.WebPages.IDF_B2.CosSim*, which only uses a different logarithm base. Approach *TF_B.VirtualDoc.IDF_C.Inner* is closest to Whitman and Lawrence [2002] and Whitman [2005]. However, Whitman et al.'s approach seems not easily describable within our framework.

4.2. Evaluation Experiments

Experiments are performed on two sets of artists. The first set (C323a) consists of 323 names of artists from 18 genres drawn from allmusic.com that are assumed to be among the best-known artists in their respective genre. From each genre, approximately the same number of artists was manually selected.

The second set (C3000a), which is more than nine times as large as the first set, comprises 3,000 artist names selected from the music information systems last.fm. We used last.fm's Web API to gather the most popular artists for each country of the world, which we then aggregated into a single list of 201,135 artist names. Since last.fm's data is prone to misspellings or other mistakes due to its collaborative, user-generated knowledge base, 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 artist names that also occur in allmusic.com, we retrieved in total 3,000 artists. This number of artists represents the typical size of a current private music collection. Both artist sets are publicly available.[12]

Please note that artist-related Web pages, which constitute the corpus, were determined using the approach presented in the last two paragraphs of Section 3.1.

It is assumed that the best performing $tf \cdot idf$ approaches will do well on both sets. This results in two stages of experiments. In the first stage, all variants are evaluated

[7]http://dir.yahoo.com/Entertainment/Music/Genres.
[8]http://www.wikipedia.org.
[9]http://www.allmusic.com.
[10]http://www.cp.jku.at/people/schedl/music/index_terms_1379.txt.
[11]The paper does not state clearly whether IDF calculation is performed on virtual documents or on individual Web pages.
[12]The first one can be downloaded from http://www.cp.jku.at/people/schedl/music/C323a.txt, the second one is available at http://www.cp.jku.at/people/schedl/music/C3000a.txt.

on the first set. Only the algorithm variants found to perform best in these experiments on the 323 artist set are then evaluated on the larger set in the second stage.

Both sets of artists are divided into the same genre categories, but have different class distributions (the number of artists of the two sets in each genre is given in parentheses): avant garde (19/8), blues (20/11), celtic (12/5), classical (17/42), country (15/24), easy listening (18/6), electronica (18/149), folk (19/24), gospel (18/23), jazz (19/106), latin (15/91), new age (17/18), rnb (20/101), rap (20/203), reggae (20/29), rock (20/2031), vocal (19/30), world (17/99).

Not wanting to go too much into detail at this point, the best-performing combination on the 323-artist-collection was *numPagesAbs.TF_C3.VirtualDoc.IDF_H.CosSim*, the combination that ranked highest on the 3,000-artist-set was *mean.TF_F. VirtualDoc.IDF_B2.Jeff*.

4.2.1. First Stage: Evaluation on the 323-Artist-Set. We model the experiments as a retrieval task. In some major aspects, we follow Buckley and Voorhees [2000] and Sanderson and Zobel [2005]. Given a query artist, the task is to find artists of the same genre via similarity. We use Mean Average Precision (MAP) as the basic performance measure. Average precision is defined as "the mean of the precision scores obtained after each relevant document is retrieved, using zero as the precision for relevant documents that are not retrieved." [Buckley and Voorhees 2000]. Following Sanderson and Zobel [2005], we first calculate MAP of each distinct algorithm variant. These are 9,248 variants. Variants that fulfill both of the following two conditions are discarded.

(1) There is a relative MAP difference of 10% or more to the top-ranked variant,
(2) and the t-test shows a significant difference to the top-ranked variant.

When doing so, and subsequently ranking all 9,248 variants according to MAP, the top 123 variants have a relative MAP difference (from the 1^{st} to the respective rank) of less than 10%. A pairwise t-test shows a significant difference for all variants except for the topmost 134 variants and the 136^{th} ranked variant. This sharp cutoff of nonsignificant vs. significant results and the relatively high accordance of our two criteria (less than 10% MAP difference and significance) supports our reasoning that these top-ranked algorithms are those worth further examination. A detailed list of the MAP scores for the best- and worst-performing variants is given in Table V.

As for the *BM25* weighting, that is, the combination of TF_G (cf. Table II) and IDF_J (cf. Table III), variant *numpagesrel.none.TF_G.wp.IDF_J.COSSIM* as best-performing combination is ranked at position 141, therefore slightly below the threshold for the MAP difference of 10% to the top-ranked variant. Although this variant is hence not included for the second stage of experiments, it is noteworthy that the *BM25* measure works best when calculating the IDF values on the level of individual Web pages, instead of modeling virtual documents.

To get more insight into which components are of high value, we look at each of the algorithm's components separately, and examine which approaches appear in the 135 selected algorithms, and how often they appear. First, it becomes apparent that only variants based on *un*normalized Web page lengths appear in the top-ranked variants. Thus, normalization does not seem to improve performance. Also, only idf computation approaches based on virtual documents are encountered. Therefore, calculating the inverse document frequency on Web pages instead of artist level in general seems not beneficial.

Figures 1 to 4 show histograms of the remaining algorithm components (page aggregation function, TF method, IDF method, and similarity measure). Note that weak performing variants have been omitted, as already described. The figures give a first insight into the relative performance of the different variants. The algorithm

Table V. MAP Scores of the Top-Ranked Variants (Notation as Described in Section 4.1.5).

MAP	Variant
0.38732	numpagesabs.none.TF_C3.vd.IDF_H.COSSIM
0.38642	numpagesabs.none.TF_C3.vd.IDF_I.COSSIM
0.38624	numpagesabs.none.TF_C2.vd.IDF_H.COSSIM
0.38523	numpagesabs.none.TF_C2.vd.IDF_I.COSSIM
0.37855	numpagesrel.none.TF_F.vd.IDF_H.COSSIM
0.37854	numpagesrel.none.TF_F.vd.IDF_I.COSSIM
0.37788	numpagesabs.none.TF_C.vd.IDF_H.COSSIM
0.37780	numpagesabs.none.TF_C.vd.IDF_I.COSSIM
0.37728	numpagesrel.none.TF_F.vd.IDF_B2.COSSIM
0.37692	mean.none.TF_F.vd.IDF_E.JEFF
0.37446	mean.none.TF_C2.vd.IDF_E.JEFF
0.37302	sum.none.TF_C2.vd.IDF_B2.COSSIM
0.37299	sum.none.TF_C2.vd.IDF_B2.JACC
0.37299	sum.none.TF_C2.vd.IDF_B2.DICE
0.37076	sum.none.TF_C3.vd.IDF_B2.COSSIM
0.37059	sum.none.TF_C3.vd.IDF_B2.JACC
0.37059	sum.none.TF_C3.vd.IDF_B2.DICE
0.37050	mean.none.TF_F.vd.IDF_B2.JEFF
0.36918	numpagesrel.none.TF_C2.vd.IDF_B2.COSSIM
0.36896	numpagesrel.none.TF_C2.vd.IDF_H.COSSIM
0.36895	numpagesrel.none.TF_C2.vd.IDF_I.COSSIM
0.36806	numpagesrel.none.TF_F.vd.IDF_I.JACC
0.36806	numpagesrel.none.TF_F.vd.IDF_I.DICE
0.36805	numpagesrel.none.TF_F.vd.IDF_H.JACC
0.36805	numpagesrel.none.TF_F.vd.IDF_H.DICE
0.36758	numpagesabs.none.TF_C2.vd.IDF_H.JACC
0.36758	numpagesabs.none.TF_C2.vd.IDF_H.DICE
0.36685	numpagesabs.none.TF_C2.vd.IDF_I.JACC
0.36685	numpagesabs.none.TF_C2.vd.IDF_I.DICE
0.36629	sum.none.TF_C2.vd.IDF_I.COSSIM
...	...
0.01097	mean.none.TF_B.vd.IDF_F.OVER
0.01097	mean.none.TF_D.vd.IDF_F.OVER
0.01081	mean.none.TF_B.vd.IDF_C.OVER
0.01075	mean.none.TF_B.vd.IDF_B.OVER
0.01075	mean.none.TF_B.vd.IDF_D.OVER
0.01055	mean.none.TF_B.vd.IDF_G.OVER
0.01044	mean.sum.TF_B.wp.IDF_F.OVER
0.01015	mean.sum.TF_B.vd.IDF_F.OVER
0.00952	mean.sum.TF_B.wp.IDF_A.OVER
0.00952	mean.sum.TF_B.vd.IDF_A.OVER

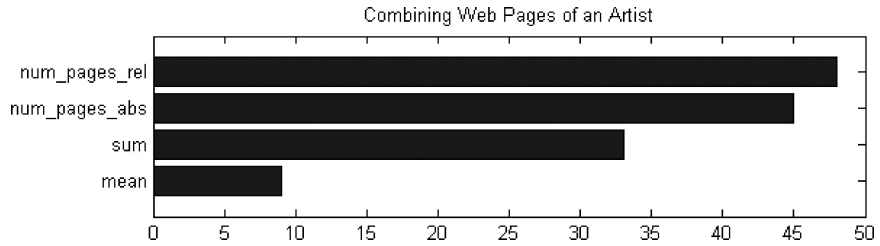

Fig. 1. Methods to combine terms appearing on an artist's Web pages. Only those appearing in the 135 selected top algorithms are shown, and the number of times they appear (totaling 135).

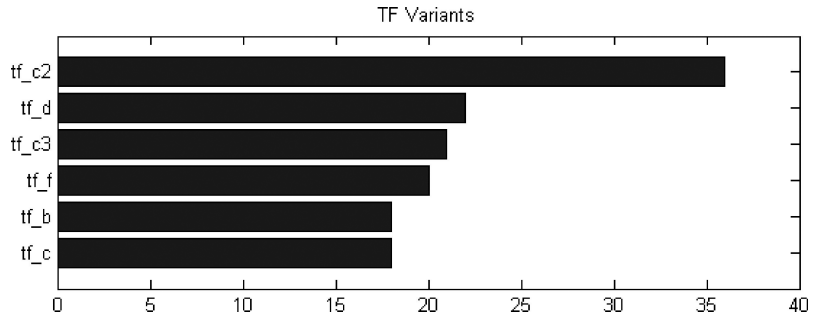

Fig. 2. TF approaches appearing in the 135 selected top algorithms, and the number of times they appear (totaling 135).

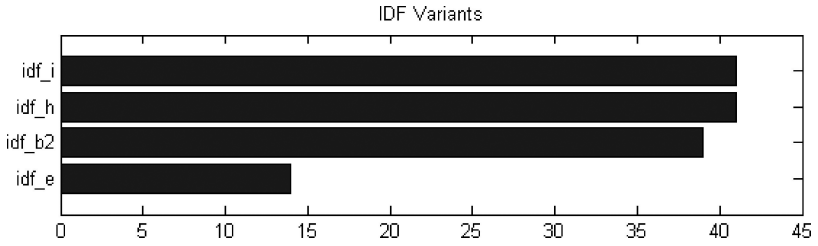

Fig. 3. IDF variants appearing in the 135 selected top algorithms, and the number of times they appear (totaling 135).

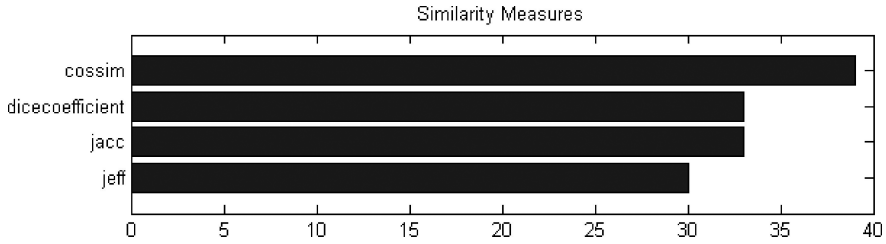

Fig. 4. Similarity measures appearing in the 135 selected top algorithms, and the number of times they appear (totaling 135).

representing the most frequently appearing variant of each component (i.e., numPages-Rel – TF_C2 – IDF_I – CosineSimilarity) is interestingly ranked only 21^{th} in the overall ranking.

However, it cannot be assumed that the shown frequencies are mutually independent. For example, when for one component of the algorithm two highly similar variants are evaluated, the other components that perform well in combination with these variants will appear more frequently.

Hence, instead of analyzing the figures more deeply, we go on by evaluating on the second set consisting of 3,000 artists all possible algorithms that can be created with the remaining variants. Thus, the only assumption is that variants that do not appear in the set of the 135 selected algorithms are not well suited for our desired algorithm to compute artist similarity. In detail, additionally to normalizing Web page length and calculating document frequency on the Web page level, the variants that are discarded here are as follows.

—*Document modeling. max*, that is, taking the maximum number of appearances over all Web pages of an artist.
—*tf computation.* variant *A* (*binary match*, i.e., if a term is contained in a document or not) and variant *E* ("alternative normalized formulation").
—*idf computation.* variants *A, B, C, D, F, G* (cf. Table III).
—*Similarity measure.* variant *INNER* (inner product), *INNER_ALT* (alternative inner product), *OVERLAP* (overlap formulation), *EUCL* (Euclidean similarity).

The remaining variants can be used to create 384 different combinations of $tf \cdot idf$ approaches and similarity measures.

4.2.2. Second Stage: Evaluation on the 3,000-Artist-Set. Since we further aim at evaluating the various approaches on a real-world collection, we retrieved the most popular artists as of the end of February 2010 from last.fm, as previously described.

In the second stage of the evaluation experiments, this 3,000-artist-set is used to investigate if both artist sets yield a comparable ranking of the 384 algorithms of interest, and which of these algorithms are top-ranked on both sets of artists. To clarify the first aspect, Spearman's rank-order correlation coefficient [Sheskin 2004] is computed on the two rankings obtained with the two artist sets. This experiment shows a correlation coefficient of 0.91. This high correlation indicates that, in general, the ranking of the algorithms is not largely influenced by factors such as size of artist collection and number of artists per genre. We note, however, that both artist sets contain mainly popular artists.

To get insight into which out of the 384 algorithms are top-ranked on both sets of artists, a ranked list of the best performing algorithms is created. In this list, algorithms are sorted based on their maximum (i.e., lowest numeric) rank in either of the two experiments (the two artist sets). For example, if an algorithm ranked second in the algorithm ranking based on the set of 323 artists, and 15^{th} on the set of 3,000 artists, then the value associated with this algorithm is 15. The corresponding list is given in the appendix.

As can be seen from the list, the $tf \cdot idf$ algorithm used in Baumann and Hummel [2003], applied to our data sets, has a maximum rank of 319. The algorithm from Knees et al. [2004] does not appear in the list, as it uses the number of Web pages to determine the document frequency, which was outside the significance bounds in the first stage. However, approach *sum.TF_C3.IDF_B2.CosSim*, which has a maximum rank of 17 in the two experiments, resembles this algorithm (counting the number of artists instead of counting the number of Web pages a term appears on). This may be seen as an indication that this variant is a good choice for the considered area of application, and it also shows that changing only one factor can have an important impact on the performance of an algorithm. Based on the latter observation, it seems that no valid statement about the relative performance of the algorithms used in Whitman and Lawrence [2002] and Whitman [2005] can be made, as the exact similarity measure used there was not evaluated in our experiments.

To gain better insights into the distribution of the different variants for the decisions regarding the algorithmic components, we show the occurrences of each algorithm variant among the various ranks (from 1 to 384). Instead of showing binary values (i.e., black for occurring/white for not occurring), for (assumed) better visibility we smoothed values by kernel density estimation. The results are reported in Figures 5 (for different aggregation functions), Figure 6 (for different term frequency formulations), Figure 7 (for different inverse document frequency), and Figure 8 (for different similarity measures). The figures' x-axes depict at which ranks the respective variants occur. Darker values indicate that the respective variant occurs more frequently in the corresponding range of ranks, while bright values indicate that the respective variant does less

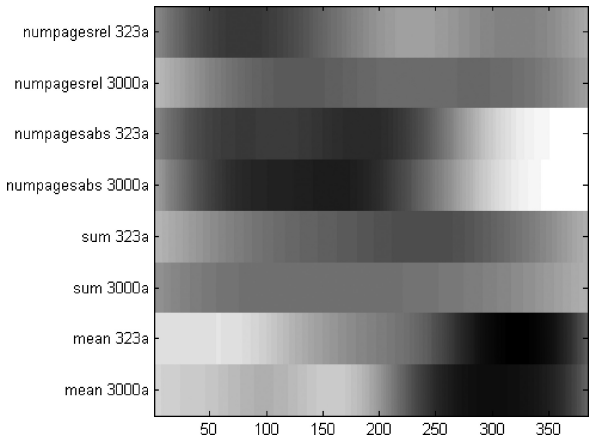

Fig. 5. Kernel density estimation for different aggregation functions.

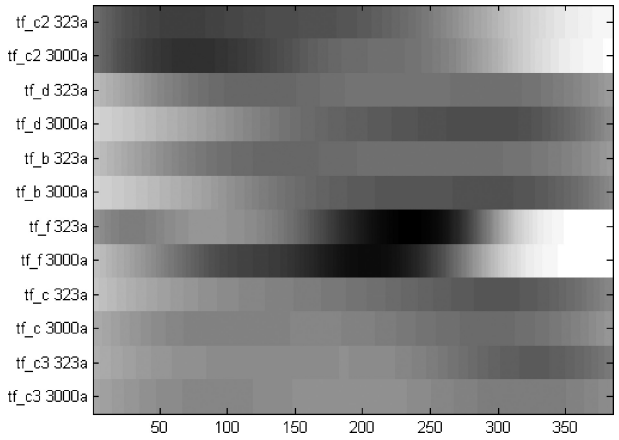

Fig. 6. Kernel density estimation for different term frequency formulations.

frequently occur in this range of ranks. From the figures, we can see a certain tendency of wide spreads in the distribution of individual variants. For example, considering Figure 7 reveals that the two best performing *idf* variants (`H` and `I`) occur in a wide range of ranks. Figure 5 demonstrates that using the mean as aggregation function is a comparably bad choice (relative to the other selected variants). From Figure 6 we can see that variant `C2` for the *tf* calculation outperforms the others considerably. Looking at Figure 7 gives no clear picture as the best performing *idf* variants `H` and `I` occur among a widespread range of ranks. As for the different similarity measures (Figure 8), although the Dice and the Jaccard coefficient performed best on average, upper ranks on C323a are dominated by the cosine measure, whereas on C3000a the Jeffrey divergence appears most frequently. In general, we can see that the cosine measure yields the most stable results, which means that the overall performance of a music similarity measure is least influenced when using the cosine measure.

5. CONCLUSIONS AND FUTURE WORK

Relative to our evaluation setting, the conclusions for calculating artist similarity can be summarized as follows. A minor finding is that normalization of each Web page

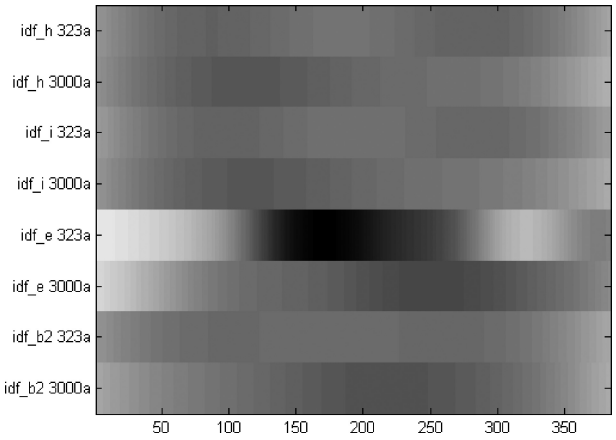

Fig. 7. Kernel density estimation for different inverse document frequency formulations.

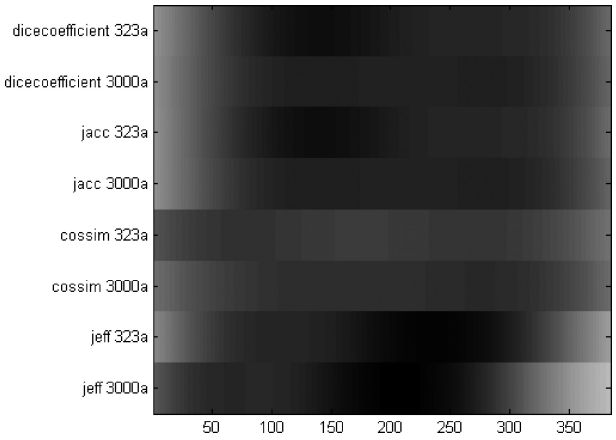

Fig. 8. Kernel density estimation for different similarity measures.

(so that each Web page has the same total weight) showed not to be of benefit. It seems, however, much more important that the document frequency for calculating *idf* is determined on virtual documents rather than on individual Web pages. Additionally, a number of possible variants did not appear in the top-ranked algorithms in the first stage of our experiments, conducted on the C323a set. Assuming that the best results are obtained when using the remaining variants, it is possible to prune the space of possible algorithms from 9,248 to 384 candidate algorithms. The frequently used cosine similarity measure appears for many of these top-ranked algorithms. However, while it was the measure in the highest ranked algorithm on the 323 artists set (*numPagesAbs.TF_C3.VirtualDoc.IDF_H.CosSim*), the algorithm that ranked highest on the 3,000-artist-set was *mean.TF_F.VirtualDoc.IDF_B2.Jeff*. Factors concerning the collection, such as size of the collection and number of artists per genre, seem to have only a minor impact on the relative performance of the best algorithms, as far as can be concluded from the evaluated parameter ranges. In contrast, a small change to an algorithm (document frequency calculated on Web pages vs. on artist level) can have an important impact on the algorithm's relative performance. The latter observation

encourages further evaluation of different text processing approaches, different term sets for indexing, term selection and term weighting functions.

On the other hand, in accordance with Zobel and Moffat [1998], we have to admit that we were not able to distill a specific combination out of the remaining 384 algorithms that worked best for both test collections, neither can we report on a choice for individual aspects (e.g., variant of term frequency, variant of similarity measure) that always outperformed all other variants. The interdependencies between different decisions which variants to choose for the individual components seem to be too large to obtain an overall winning combination. Thus, Zobel and Moffat's final statement, "The measures do not form a space that can be explored in any meaningful way, other than by exhaustion," does unfortunately also apply analogously to the music similarity space derived from music-related Web pages. But considering that we are able to restrict this space to 384 candidate algorithms in our evaluation setting, exhaustion within this subspace seems feasible.

This study focused on the task of (text-based) similarity estimation between music artists, which is a relatively specific, nevertheless important, task in music information research. Other MIR tasks such as artist clustering, text-based music retrieval, or automated playlist generation might require other formulations of algorithm variants. It seems reasonable to conclude that, depending on the task, various parameter choices need to be evaluated. Nevertheless, the results of this study may support research towards personalized music retrieval as well as combining different aspects of music similarity. For example, Zhang et al. [2009] propose a system for personalized music search, taking into account similarity aspects derived from music content and from social factors. A multimodal music similarity model taking subjective aspects into account is also presented in McFee and Lanckriet [2009]. Since such work on personalized MIR systems is strongly related to text-based representation of (music-related) documents—not only of artist pages, but also of user-generated content (e.g., instant messages or social network posts)—efficient term weighting and similarity measures are crucial. Furthermore, approaches that combine content-based with context-based information for the purpose of music playlist generation, such as Pohle et al. [2007], are likely to benefit from the results of this study.

As for future work, the current experiments are limited to the rather narrow task of genre classification. The genre assignment of the two test collections used originates from allmusic.com's experts' judgments. A possibly more accurate ground truth could be derived from "similar artist"-relations given by last.fm's collaborative filtering approach. Even though this data is likely prone to a population bias and information may be sparse [Schedl and Knees 2009], evaluation against such a ground truth definition may yield interesting findings.

Another direction in which to extend the work at hand is determining the influence of individual choices made in the analyzed variants for normalization, aggregation, *tf* and *idf* formulations, and similarity measurement. To this end, a general linear regression model could be used to assess the relative impact of various decisions.

APPENDIX : DETAILED RESULTS

In the following, a sorted list of the best performing approaches is given. The number gives the lower of the two ranks (obtained on the 323-artist-set and the 3,000-artist-set). The list contains all combinations that differed not significantly from the respective best variant – neither on the 323-artist-set, nor on the 3,000-artist-set. Entries have the form `<PageAggregationFunction>.<TF-Approach>.<IDF-Approach>.<SimilarityMeasure>`.

6. numpagesabs.tf_c3.idf_i.cossim
8. numpagesabs.tf_c3.idf_h.cossim
9. numpagesabs.tf_c2.idf_i.cossim
10. numpagesabs.tf_c2.idf_h.cossim
13. mean.tf_f.idf_e.jeff
15. sum.tf_c2.idf_b2.cossim
16. mean.tf_f.idf_b2.jeff
17. sum.tf_c3.idf_b2.cossim
18. sum.tf_c2.idf_b2.dice
19. sum.tf_c2.idf_b2.jacc
22. numpagesabs.tf_c.idf_i.cossim
23. numpagesabs.tf_c.idf_h.cossim
31. sum.tf_c2.idf_i.cossim
32. sum.tf_c2.idf_h.cossim
35. sum.tf_c3.idf_b2.dice
35. sum.tf_c3.idf_i.cossim
36. sum.tf_c3.idf_b2.jacc
36. numpagesrel.tf_c2.idf_b2.jeff
37. numpagesabs.tf_b.idf_b2.jeff
38. numpagesabs.tf_d.idf_b2.jeff
39. numpagesrel.tf_b.idf_b2.jeff
40. numpagesrel.tf_d.idf_b2.jeff
41. mean.tf_c2.idf_b2.jeff
44. sum.tf_c3.idf_h.cossim
47. numpagesrel.tf_f.idf_b2.jeff
48. sum.tf_c.idf_b2.cossim
51. sum.tf_c.idf_i.cossim
55. sum.tf_c.idf_b2.dice
56. sum.tf_c.idf_b2.jacc
57. numpagesabs.tf_c2.idf_h.dice
58. numpagesabs.tf_c2.idf_h.jacc
59. numpagesabs.tf_c3.idf_b2.cossim
60. numpagesabs.tf_c2.idf_i.dice
61. numpagesabs.tf_c2.idf_i.jacc
62. sum.tf_c.idf_h.cossim
62. numpagesabs.tf_c3.idf_h.dice
63. numpagesabs.tf_c3.idf_h.jacc
64. numpagesabs.tf_c3.idf_i.dice
65. numpagesabs.tf_c3.idf_i.jacc
68. numpagesrel.tf_f.idf_i.cossim
69. numpagesrel.tf_f.idf_h.cossim
70. numpagesabs.tf_c.idf_h.dice
71. numpagesabs.tf_c.idf_h.jacc
71. numpagesabs.tf_c2.idf_b2.cossim
74. numpagesabs.tf_c.idf_i.dice
75. numpagesabs.tf_c.idf_i.jacc
75. numpagesrel.tf_b.idf_i.jeff
76. numpagesrel.tf_d.idf_i.jeff
77. numpagesrel.tf_b.idf_h.jeff
78. numpagesabs.tf_c3.idf_b2.dice
78. numpagesrel.tf_c2.idf_i.jeff
79. numpagesabs.tf_c3.idf_b2.jacc
79. numpagesrel.tf_d.idf_h.jeff
80. numpagesrel.tf_c2.idf_h.jeff
81. numpagesabs.tf_b.idf_h.jeff
82. mean.tf_c2.idf_e.jeff
82. numpagesabs.tf_d.idf_h.jeff
85. numpagesrel.tf_f.idf_h.jeff
86. numpagesrel.tf_c2.idf_e.jeff
86. numpagesrel.tf_f.idf_i.jeff
87. numpagesrel.tf_f.idf_e.jeff
88. numpagesabs.tf_b.idf_i.jeff
89. numpagesabs.tf_b.idf_e.jeff
89. numpagesabs.tf_d.idf_i.jeff
90. mean.tf_f.idf_h.jeff
90. numpagesabs.tf_d.idf_e.jeff
91. sum.tf_c2.idf_h.dice
91. numpagesrel.tf_b.idf_e.jeff
92. sum.tf_c2.idf_h.jacc
92. numpagesrel.tf_d.idf_e.jeff
93. sum.tf_c2.idf_i.dice
93. numpagesrel.tf_f.idf_h.dice
94. sum.tf_c2.idf_i.jacc
94. numpagesrel.tf_f.idf_h.jacc
95. numpagesabs.tf_c2.idf_b2.dice
95. numpagesrel.tf_f.idf_i.dice
96. numpagesabs.tf_c2.idf_b2.jacc
96. numpagesrel.tf_f.idf_i.jacc
97. numpagesrel.tf_c2.idf_h.cossim
98. sum.tf_c3.idf_h.dice
98. numpagesrel.tf_c2.idf_i.cossim
99. sum.tf_c3.idf_h.jacc
99. numpagesrel.tf_f.idf_b2.cossim
100. sum.tf_c3.idf_i.dice
100. numpagesrel.tf_c2.idf_h.dice
101. sum.tf_c3.idf_i.jacc
101. numpagesrel.tf_c2.idf_h.jacc
102. sum.tf_c.idf_h.dice
102. numpagesrel.tf_c2.idf_i.dice
103. sum.tf_c.idf_h.jacc
103. numpagesrel.tf_c2.idf_i.jacc
104. mean.tf_f.idf_i.jeff
104. numpagesabs.tf_b.idf_h.cossim
105. numpagesabs.tf_c.idf_b2.cossim
105. numpagesabs.tf_d.idf_h.cossim
106. sum.tf_c.idf_i.dice
106. numpagesrel.tf_b.idf_h.cossim
107. sum.tf_c.idf_i.jacc
107. numpagesrel.tf_d.idf_h.cossim
108. sum.tf_c3.idf_e.dice
108. numpagesrel.tf_b.idf_i.cossim
109. sum.tf_c3.idf_e.jacc
109. numpagesrel.tf_d.idf_i.cossim
110. mean.tf_f.idf_b2.cossim
110. sum.tf_c2.idf_e.dice
111. sum.tf_c2.idf_e.jacc
111. numpagesabs.tf_b.idf_i.cossim
112. numpagesabs.tf_c.idf_b2.dice
112. numpagesabs.tf_d.idf_i.cossim
113. numpagesabs.tf_c.idf_b2.jacc
113. numpagesrel.tf_c2.idf_b2.cossim
114. sum.tf_c2.idf_e.cossim
114. numpagesrel.tf_f.idf_b2.dice
115. sum.tf_c3.idf_e.cossim
115. numpagesrel.tf_f.idf_b2.jacc

All variants do not use page length normalization, and the document frequency is always calculated on virtual documents (and not taken as the number of Web pages). For brevity, these choices are not mentioned explicitly.

REFERENCES

AHN, L. V. AND DABBISH, L. 2004. Labeling images with a computer game. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

AUCOUTURIER, J.-J. AND PACHET, F. 2002. Scaling up music playlist generation. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'02)*. 105–108.

AUCOUTURIER, J.-J. AND PACHET, F. 2004. Improving timbre similarity: How high is the sky? *J. Neg. Results Speech Audio Sci. 1,* 1.

BACCIGALUPO, C., PLAZA, E., AND DONALDSON, J. 2008. Uncovering affinity of artists to multiple genres from social behaviour data. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*.

BAEZA-YATES, R. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*. Addison Wesley.

BAUMANN, S. AND HUMMEL, O. 2003. Using cultural metadata for artist recommendation. In *Proceedings of the Conference on Web Delivering of Music (WEDELMUSIC'02)*.

BERENZWEIG, A., LOGAN, B., ELLIS, D. P., AND WHITMAN, B. 2003. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*.

BUCKLEY, C. AND VOORHEES, E. 2000. Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

CASEY, M. A., VELTKAMP, R., GOTO, M., LEMAN, M., RHODES, C., AND SLANEY, M. 2008. Content-based music information retrieval: Current directions and future challenges. *Proc. IEEE 96*, 668–696.

CELMA, O., CANO, P., AND HERRERA, P. 2006. Search sounds: An audio crawler focused on weblogs. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*.

CELMA, O. AND LAMERE, P. 2007. ISMIR 2007 Tutorial: Music recommendation. http://mtg.upf.edu/~ocelma/MusicRecommendationTutorial-ISMIR2007 (last accessed 12/07).

CHAKRABARTI, S., VAN DEN BERG, M., AND DOM, B. 1999. Focused crawling: A new approach to topic-specific web resource discovery. *Comput. Netw. 31,* 11–16, 1623–1640.

CIMIANO, P., HANDSCHUH, S., AND STAAB, S. 2004. Towards the self-annotating Web. In *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*. ACM Press, New York, NY, 462–471.

CIMIANO, P. AND STAAB, S. 2004. Learning by Googling. *ACM SIGKDD Explor. Newsle. 6,* 2, 24–33.

COHEN, W. W. AND FAN, W. 2000a. Web-collaborative filtering: Recommending music by crawling the Web. *Comput. Netw. 33,* 1–6, 685–698.

DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. 1990. Indexing by latent semantic analysis. *J. Am. Soc. Inform. Sci. 41*, 391–407.

DOWNIE, J. S. 2003. Toward the scientific evaluation of music information retrieval systems. In *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR'03)*.

ELLIS, D. P. W. 2002. The quest for ground truth in musical artist similarity. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*.

FINGERHUT, M. 2004. Music information retrieval, or how to search for (and maybe find) music and do away with incipits. Slides for IAML/IASA Congress.

GELEIJNSE, G. AND KORST, J. 2006. Web-based artist categorization. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*.

GÖKER, A. AND MYRHAUG, H. I. 2002. User context and personalisation. In *Proceedings of the 6th European Conference on Case Based Reasoning (ECCBR'02)* (Workshop on Case Based Reasoning and Personalization).

GOVAERTS, S. AND DUVAL, E. 2009. A Web-based approach to determine the origin of an artist. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR'09)*.

HOFMANN, T. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

HU, X., DOWNIE, J. S., AND EHMANN, A. F. 2009. Lyric text mining in music mood classification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR'09)*.

HU, X., DOWNIE, J. S., WEST, K., AND EHMANN, A. 2005. Mining music reviews: Promising preliminary results. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*.

KASSLER, M. 1966. Musical information retrieval. *Perspect. New Music* 4, 59–67.

KNEES, P., PAMPALK, E., AND WIDMER, G. 2004. Artist classification with Web-based data. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR'04)*. 517–524.

KNEES, P., POHLE, T., SCHEDL, M., AND WIDMER, G. 2007. 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'07)*.

KNEES, P., SCHEDL, M., AND POHLE, T. 2008. A deeper look into Web-based classification of music artists. In *Proceedings of the 2nd Workshop on Learning the Semantics of Audio Signals (LSAS'08)*.

KNEES, P., SCHEDL, M., POHLE, T., AND WIDMER, G. 2007. Exploring music collections in virtual landscapes. *IEEE MultiMed. 14,* 3, 46–54.

LAURIER, C., GRIVOLLA, J., AND HERRERA, P. 2008. Multimodal music mood classification using audio and lyrics. In *Proceedings of the International Conference on Machine Learning and Applications*.

LAW, E. L. M., VON AHN, L., DANNENBERG, R. B., AND CRAWFORD, M. 2007. Tagatune: A game for music and sound annotation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

LOGAN, B., ELLIS, D. P. W., AND BERENZWEIG, A. 2003. Toward evaluation techniques for music similarity. In *Proceedings of the Workshop on the Evaluation of Music Information Retrieval (MIR) Systems at SIGIR*.

LOGAN, B., KOSITSKY, A., AND MORENO, P. 2004. Semantic Analysis of Song Lyrics. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'04)*.

MAHEDERO, J. P. G., MARTÍNEZ, A., CANO, P., KOPPENBERGER, M., AND GOUYON, F. 2005. Natural language processing of lyrics. In *Proceedings of the 13th ACM International Conference on Multimedia (MM'05)*. 475–478.

MANDEL, M. I. AND ELLIS, D. P. W. 2007. A web-based game for collecting music metadata. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

MCFEE, B. AND LANCKRIET, G. 2009. Heterogeneous embedding for subjective artist similarity. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR'09)*.

PACHET, F. AND CAZALY, D. 2000. A taxonomy of musical genre. In *Proceedings of Content-Based Multimedia Information Access (RIAO) Conference*.

PACHET, F., WESTERMANN, G., AND LAIGRE, D. 2001. Musical data mining for electronic music distribution. In *Proceedings of the 1st International Conference on WEB Delivering of Music (WEDELMUSIC'01)*.

PAMPALK, E. 2006. Computational models of music similarity and their application to music information retrieval. Ph.D. thesis, Vienna University of Technology.

PAMPALK, E., FLEXER, A., AND WIDMER, G. 2005. Hierarchical organization and description of music collections at the artist level. In *Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'05)*.

PAMPALK, E. AND GOTO, M. 2007. MusicSun: A new approach to artist recommendation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

PAMPALK, E., RAUBER, A., AND MERKL, D. 2002. Content-based organization and visualization of music archives. In *Proceedings of the 10th ACM International Conference on Multimedia (MM'02)*. 570–579.

PÉREZ-IGLESIAS, J., PÉREZ-AGÜERA, J. R., FRESNO, V., AND FEINSTEIN, Y. Z. 2009. Integrating the probabilistic models BM25/BM25F into Lucene. CoRR abs/0911.5046.

POHLE, T. 2009. Automatic characterization of music for intuitive retrieval. Ph.D. thesis, Johannes Kepler University Linz, Austria.

POHLE, T., KNEES, P., SCHEDL, M., PAMPALK, E., AND WIDMER, G. 2007c. "Reinventing the Wheel": A novel approach to music player interfaces. *IEEE Trans. Multimed. 9*, 567–575.

POHLE, T., KNEES, P., SCHEDL, M., AND WIDMER, G. 2007a. Building an interactive next-generation artist recommender based on automatically derived high-level concepts. In *Proceedings of the 5th International Workshop on Content Based Multimedia Indexing (CBMI'07)*.

POHLE, T., KNEES, P., SCHEDL, M., AND WIDMER, G. 2007b. Meaningfully browsing music services. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

ROBERTSON, S., WALKER, S., AND BEAULIEU, M. 1999. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track. In *Proceedings of the 7th Text REtrieval Conference*. 253–264.

ROBERTSON, S., WALKER, S., AND HANCOCK-BEAULIEU, M. 1995. Large test collection experiments on an operational, interactive system: Okapi at TREC. *In Inform. Process. Manage. 31*, 345–360.

THORPE, S., FIZE, D. , AND MARLOT, C. 1996. Speed of processing in the human visual system. *Nature 381,* 6582, 520–522.

SANDERSON, M. AND ZOBEL, J. 2005. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*.

SCHEDL, M. 2008. Automatically extracting, analyzing, and visualizing information on music artists from the World Wide Web. Ph.D. thesis, Johannes Kepler University Linz, Austria.

SCHEDL, M. AND KNEES, P. 2009. Context-based music similarity estimation. In *Proceedings of the 3rd International Workshop on Learning the Semantics of Audio Signals (LSAS'09)*.

SCHEDL, M., PAMPALK, E., AND WIDMER, G. 2005. Intelligent structuring and exploration of digital music collections. *e&i—Elektrotechnik und Informationstechnik 122,* 7–8, 232–237.

SCHEDL, M. AND POHLE, T. 2010. Enlightening the sun: A user interface to explore music artists via multimedia content. *Multimed. Tools Appl. 49*, 1, (Special Issue on Semantic and Digital Media Technologies) 101–118.

SCHEDL, M., SEYERLEHNER, K., WIDMER, G., AND SCHIKETANZ, C. 2010. Three Web-based heuristics to determine a person's or institution's country of origin. In *Proceedings of the 33th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*.

SCHEDL, M. AND WIDMER, G. 2007. Automatically detecting members and instrumentation of music bands via web content mining. In *Proceedings of the 5th Workshop on Adaptive Multimedia Retrieval (AMR'07)*.

SCHEDL, M., WIDMER, G., POHLE, T., AND SEYERLEHNER, K. 2007. Web-based detection of music band members and line-up. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

SCHNITZER, D., POHLE, T., KNEES, P., AND WIDMER, G. 2007. One-touch access to music on mobile devices. In *Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia (MUM'07)*.

SEYERLEHNER, K., POHLE, T., SCHEDL, M., AND WIDMER, G. 2007. Automatic music detection in television productions. In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx'07)*.

SHAVITT, Y. AND WEINSBERG, U. 2009. Songs clustering using peer-to-peer co-occurrences. In *Proceedings of the IEEE International Symposium on Multimedia (ISM'09): International Workshop on Advances in Music Information Research (AdMIRe'09)*.

SHESKIN, D. J. 2004. *Handbook of Parametric and Nonparametric Statistical Procedures*, 3rd Ed. Chapman & Hall/CRC, Boca Raton.

STENZEL, R. AND KAMPS, T. 2005. Improving content-based similarity measures by training a collaborative model. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*.

TURNBULL, D., BARRINGTON, L., AND LANCKRIET, G. 2008. Five approaches to collecting tags for music. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*.

TURNBULL, D., LIU, R., BARRINGTON, L., AND LANCKRIET, G. 2007. A game-based approach for collecting semantic annotations of music. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

WHITMAN, B. 2005. Learning the meaning of music. Ph.D. thesis, School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA.

WHITMAN, B. AND LAWRENCE, S. 2002. Inferring descriptions and similarity for music from community metadata. In *Proceedings of the International Computer Music Conference (ICMC)*. 591–598

ZADEL, M. AND FUJINAGA, I. 2004. Web services for music information retrieval. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR'04)*.

ZHANG, B., SHEN, J., XIANG, Q., AND WANG, Y. 2009. CompositeMap: A novel framework for music similarity measure. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, New York, NY, 403–410.

ZOBEL, J. AND MOFFAT, A. 1998. Exploring the similarity space. *ACM SIGIR Forum 32,* 1, 18–34.

ZOBEL, J. AND MOFFAT, A. 2006. Inverted files for text search engines. *ACM Comput. Surv. 38*, 1–56.

Received May 2010; revised November 2010; accepted January 2011