Information Processing and Management 47 (2011) 426–439

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

# A music information system automatically generated via Web content mining techniques

Markus Schedl *, Gerhard Widmer, Peter Knees, Tim Pohle

*Department of Computational Perception, Johannes Kepler University, Altenberger Straße 69, A-4040 Linz, Austria*

ARTICLE INFO

*Article history:*
Received 2 April 2009
Received in revised form 23 August 2010
Accepted 6 September 2010
Available online 8 October 2010

*Keywords:*
Music information retrieval
Web content mining
Information systems
Application
Evaluation

ABSTRACT

This article deals with the problem of *mining music-related information from the Web* and representing this information via a *music information system*. Novel techniques have been developed as well as existing ones refined in order to automatically gather information about music artists and bands. After searching, retrieval, and indexing of Web pages that are related to a music artist or band, *Web content mining* and *music information retrieval* techniques were applied to capture the following categories of information: *similarities between music artists or bands*, *prototypicality of an artist or a band for a genre*, *descriptive properties of an artist or a band*, *band members and instrumentation*, *images of album cover artwork*. Approaches to extracting these pieces of information are presented and evaluation experiments are described that investigate the proposed approaches' performance. From the insights gained by the various experiments an *Automatically Generated Music Information System* (AGMIS) providing Web-based access to the extracted information has been developed. AGMIS demonstrates the feasibility of automated music information systems on a large collection of more than 600,000 music artists.



## 1. Introduction and context

Over the past few years, digital music distribution via the World Wide Web has seen a tremendous increase. As a result, music-related information beyond the pure digital music file (musical meta-data) is becoming more and more important as users of online music stores nowadays expect to be offered such additional information. Moreover, digital music distributors are in need of such additional value that represents a decisive advantage over their competitors.

Also music information systems, i.e., systems primarily focusing on *providing information* about music, not on selling music, typically offer multimodal information about music artists,[1] albums, and tracks (e.g., genre and style, similar artists, biographies, song samples, or images of album covers). In common music information systems, such information is usually collected and revised by experts, e.g., *All Music Guide* (amg, 2009) or relies on user participation, e.g., *last.fm* (las, 2009). In contrast, this paper describes methods for building such a system by automatically extracting the required information from the Web at large. To this end, various techniques to estimate relations between artists, to determine descriptive terms, to extract band members and instrumentation, and to find images of album covers were elaborated, evaluated, refined, and aggregated.

Automatically retrieving information about music artists is an important task in music information retrieval (MIR), cf. Downie (2003). It permits, for example, enriching music players with meta-information (Schedl, Pohle, Knees, & Widmer, 2006c), automatically tagging of artists (Eck, Bertin-Mahieux, & Lamere, 2007), automatic biography generation (Alani et al., 2003), developing user interfaces to browse music collections by more sophisticated means than the textual browsing

* Corresponding author. Tel.: +43 (0) 732 2468 1512; fax: +43 (0) 732 2468 1520.
*E-mail address:* markus.schedl@jku.at (M. Schedl).

[1] In the following, we use the term "artist" to refer to both single musicians and bands.

0306-4573/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2010.09.002

facilities, in an artist − album − track hierarchy, traditionally offered (Knees, Schedl, Pohle, & Widmer, 2006; Pampalk & Goto, 2007), or defining similarity measures between artists. Music similarity measures can then be used, for example, to create relationship networks (Cano & Koppenberger, 2004), for automatic playlist generation (Aucouturier & Pachet, 2002; Pohle, Knees, Schedl, Pampalk, & Widmer, 2007), or to build music recommender systems (Celma & Lamere, 2007; Zadel & Fujinaga, 2004) or music search engines (Knees, Pohle, Schedl, & Widmer, 2007).

In the following, an overview of existing Web mining techniques for MIR is given in Section 2. Section 3 briefly presents the methods developed and refined by the authors, together with evaluation results. Section 4 describes the application of the techniques from Section 3 for creating the *Automatically Generated Music Information System* (AGMIS), a system providing information on more than 600,000 music artists. Finally, in Section 5, conclusions are drawn, and directions for future work are pointed out.

## 2. Related work

Related work mainly consists of methods to derive similarities between music artists and attribute descriptive terms to an artist, which is also known as *tagging*. Traditionally, similarities between songs or artists are calculated on some kind of musically relevant features extracted form the audio signal. Such features usually aim at capturing *rhythmic* or *timbral* aspects of music. Rhythm is typically described by some sort of *beat histogram*, e.g., Pampalk, Rauber, and Merkl (2002) and Dixon, Gouyon, and Widmer (2004 et al.), whereas timbral aspects are usually approximated by *Mel Frequency Cepstral Coefficients* (MFCCs), e.g., Aucouturier, Pachet, and Sandler (2005) and Mandel and Ellis (2005). However, such audio signal-based similarity measures cannot take into account aspects like the cultural context of an artist, the semantics of the lyrics of a song, or the emotional impact of a song on its listener. In fact, the performance of such purely audio-based measures seems to be limited by a "glass ceiling", cf. Aucouturier and Pachet (2004).

Overcoming this limitation requires alternative methods, most of which have in common the *participation of lots of people* to form a large information resource. Like typical Web 2.0 applications, such methods benefit from the wisdom of the crowd. The respective data is hence often called *cultural features* or *community meta-data*. Probably the most prominent example of such features are those gained in a collaborative tagging process. Lamere (2008) gives a comprehensive overview of the power of social tags in the music domain, shows possible applications, but also outlines shortcomings of collaborative tagging systems. Celma (2008) laboriously analyzed and compared different tagging approaches for music, especially focusing on their use for music recommendation and taking into account the long tail of largely unknown artists.

Cultural features were, however, already used in MIR before the Web 2.0-era and the emergence of folksonomies. Early approaches inferring music similarity from sources other than the audio signal use, e.g., co-occurrences of artists or tracks in radio station playlists and compilation CDs (Pachet, Westerman, & Laigre, 2001) or in arbitrary lists extracted from Web pages (Cohen & Fan, 2000). Other researchers extracted different term sets from artist-related Web pages and built individual term profiles for each artist (Ellis, Whitman, Berenzweig, & Lawrence, 2002; Knees, Pampalk, & Widmer, 2004; Whitman & Lawrence, 2002). The principal shortcoming of such similarities inferred from cultural features is their restriction to the artist level since there is usually too little data available on the level of individual songs. The most promising approach to transcend these limitations is combining multiple features extracted from different sources. For example, a method that enriches Web-based with audio-based features to create term profiles at the track level is proposed in Knees, Pohle, et al. (2007). The authors present a search engine to retrieve music by textual queries, like "rock music with great riffs". Pohle et al. (2007) present an approach to automatic playlist generation that approximates the solution to a Traveling Salesman Problem on signal-based distances, but uses Web-based similarities to direct the search heuristics.

As for determining descriptive terms for an artist, such as instruments, genres, styles, moods, emotions, or geographic locations, Pampalk, Flexer, and Widmer (2005) use a self-assembled dictionary and apply different term weighting techniques on artist-related Web pages to assign terms to sets of artists and cluster them in a hierarchical manner. The term weighting functions analyzed were based on document frequency (DF), term frequency (TF), and term frequency · inverse document frequency (TF·IDF) variations. The conducted experiments showed that considering only the terms in the dictionary outperforms using the unpruned, complete set of terms extracted from the Web pages. Geleijnse and Korst (2006) and Schedl et al. (2006c) independently present an approach to artist tagging that estimates the conditional probability for the artist name under consideration to be found on a Web page containing a specific descriptive term and the probability for the descriptive term to occur on a Web page known to mention the artist name. The calculated probabilities are used to predict the most probable value of attributes related to artist or music (e.g., *happy*, *neutral*, *sad* for the attribute *mood*). Both papers particularly try to categorize artists according to their genre, which seems reasonable as genre names are also among the most frequently applied tags in common music information systems like *last.fm* (Geleijnse, Schedl, & Knees, 2007). Another category of tagging approaches make use of *last.fm* tags and distill certain kinds of information. For example, Hu, Bay, and Downie (2007) use a part-of-speech (POS) tagger to search *last.fm* tags for adjectives that describe the mood of a song. Eck et al. (2007) use the machine learning algorithm AdaBoost to learn relations between acoustic features and *last.fm* tags.

A recent approach to gathering tags is the so-called *ESP games* (von Ahn & Dabbish, 2004). These games provide some form of incentive[2] to the human player to solve problems that are hard to solve for computers, e.g., capturing emotions evoked

[2] Commonly the pure joy of gaming is enough to attract players.

when listening to a song. Turnbull, Liu, Barrington, and Lanckriet (2007), Mandel and Ellis (2007), and Law, von Ahn, Dannenberg, and Crawford (2007) present such game-style approaches that provide a fun way to gather musical annotations.

## 3. Mining the Web for music artist-related information

All methods proposed here rely on the availability of artist-related data on the Web. The authors' principal approach to extracting such data is the following. Given only a list of artist names, we first query a search engine[3] to retrieve the URLs of up to 100 top-ranked search results for each artist. The content available at these URLs is extracted and stored for further processing. To overcome the problem of artist names that equal common speech words and to direct the search towards the desired information, we use task-specific query schemes like `"band name" + music + members` to obtain data related to band members and instrumentation. We do not account for multilingual pages by varying the language of the additional keywords (e.g., `"music"`, `"Musik"`, `"musique"`, `"musica"`) as this would considerably increase the number of queries issued to the search engine. It has to be kept in mind, however, that restricting the search space to English pages might yield undiscovered pages which are nevertheless relevant to the artist. In any case, this approach relies on the ranking algorithm of the search engine.

Depending on the task to solve, either a *document-level inverted index* or a *word-level index* (Zobel & Moffat, 2006) is then created from the retrieved Web pages. In some cases, especially when it comes to artist tagging, a special dictionary of musically relevant terms is used for indexing. After having indexed the Web pages, we gain artist-related information of various kinds as described in the following.

As an alternative approach to the use of a search engine for Web page selection, we could use a focused crawler (Chakrabarti, van den Berg, & Dom, 1999) trained to retrieve pages from the music domain. We are currently assessing this alternative as it would avoid relying on commercial search engines and would allow us to build a corpus specific to the music domain. On the other hand, companies like *Google* offer a huge corpus which can be accessed very efficiently. Thus, we still have to compare these two strategies (directed search using a search engine vs. focused crawling) and assess their performance in depth, which will be part of future work.

### 3.1. Relations between artists

#### 3.1.1. Similarity Relations

A key concept in music information retrieval and crucial part of any music information system is *similarity relations* between artists. To model such relations, we propose an approach that is based on co-occurrence analysis (Schedl, Knees, & Widmer, 2005a). More precisely, the similarity between two artists $i$ and $j$ is inferred from the conditional probability that the artist name $i$ occurs on a Web page that was returned as response to the search query for the artist name $j$ and vice versa. The formal definition of the similarity measure is given in Formula (1), where $I$ represents the set of Web pages returned for artist $i$ and $df_{i,J}$ is the document frequency of the artist name $i$ calculated on the set of Web pages returned for artist $j$.

$$sim_{cooc}(i,j) = \frac{1}{2} \cdot \left( \frac{df_{i,J}}{|J|} + \frac{df_{j,I}}{|I|} \right) \tag{1}$$

Having calculated the similarity for each pair of artists in the input list, it is possible to output, for any artist, a list of most similar artists, i.e., building a recommender system. Evaluation in an artist-to-genre classification task using a *k-nearest neighbor classifier* on a set of 224 artists from 14 genres yielded accuracy values of about 85% averaged over all genres, cf. Schedl et al. (2005a).

#### 3.1.2. Prototypicality relations

Co-occurrences of artist names on Web pages (together with genre information) can also be used to derive information about the *prototypicality of an artist for a certain genre* (Schedl, Knees, & Widmer, 2005b, 2006). To this end, the asymmetry of the one-sided, co-occurrence-based similarity measure is exploited as explained below. Taking a look at Formula (1) again and focusing on the single terms $\frac{df_{i,J}}{|J|}$ and $\frac{df_{j,I}}{|I|}$ that estimate the single probability for an artist name to be found on the page retrieved for another artist, it is obvious that, in general, $\frac{df_{i,J}}{|J|} \neq \frac{df_{j,I}}{|I|}$. Such asymmetric similarity measures have some disadvantages, the most important of which is that they do not allow to induce a metric in the feature space. Moreover, they produce unintuitive and hard to understand visualizations when using them to build visual browsing applications based on clustering, like the *nepTune* interface (Knees, Schedl, Pohle, & Widmer, 2007). However, the asymmetry can also be beneficially exploited for deriving artist popularity or prototypicality of an artist for a certain genre (or any other categorical aspect). Taking into account the asymmetry of the co-occurrence-based similarity measure, the main idea behind our approach is that it is more likely to find the name of a well-known and representative artist for a genre on many Web pages about a lesser known artist, e.g., a newcomer band, than vice versa. To formalize this idea, we developed an approach that is based on the *backlink/forward link-ratio* of two artists $i$ and $j$ from the same genre, where a *backlink* of $i$ from $j$ is defined as any occurrence of artist $i$ on a Web page that is known to contain artist $j$, whereas a *forward link* of $i$ to $j$ is defined as any

[3] We commonly used *Google* (goo, 2009), but also experimented with *exalead* (exa, 2009).

occurrence of $j$ on a Web page known to mention $i$. Relating the number of forward links to the number of backlinks for each pair of artists from the same genre, a ranking of the artist prototypicality for the genre under consideration is obtained. More precisely, we count the number of forward links and backlinks on the document frequency-level, i.e., all occurrences of artist name $i$ on a particular page retrieved for $j$ contribute 1 to the backlink count of $i$, regardless of the term $i$'s frequency on this page. To alleviate the problem of artist names being highly ranked due to their resemblance to common speech words,[4] we use a correction factor that penalizes artists whose prototypicality is exorbitantly, therefore unjustifiably, high for all genres. Putting this together, the refined prototypicality ranking function $r(i,g)$ of artist $i$ for genre $g$ is given in Formula (2), where $G$ represents the set of artists in genre $g$. The penalization term is given in Formula (3), where $A$ denotes the set of all artists in the collection. The functions $bl(i,j)$ and $fl(i,j)$ as defined in Formulas (4) and (5), respectively, measure whether the number of backlinks of $i$ from $j$, as defined above, exceeds the number of forward links of $i$ to $j$ (in this case, $bl(i,j) = 1$ and $fl(i,j) = 0$) or the number of backlinks of $i$ from $j$ is equal or less than the number of forward links of $i$ from $j$ (in this case, $bl(i,j) = 0$ and $fl(i,j) = 1$). $df_{j,I}$ gives the number of Web pages retrieved for artist $i$ that also mention artist $j$. This number hence represents a document frequency and equals the respective term in Formula (1). $|I|$ is the total number of pages retrieved for artist $i$. The normalization function $\|\cdot\|$ shifts all values to the positive range and maps them to $[0,1]$.

$$r(i,g) = \frac{\sum_{j\in G}^{j\neq i} bl(i,j)}{\sum_{j\in G}^{j\neq i} fl(i,j) + 1} \cdot penalty(i) \tag{2}$$

$$penalty(i) = \left\| \log\left( \frac{\sum_{j\in A}^{j\neq i} fl(i,j) + 1}{\sum_{j\in A}^{j\neq i} bl(i,j) + 1} \right) \right\|^2 \tag{3}$$

$$bl(i,j) = \begin{cases} 1 & \text{if } \frac{df_{j,I}}{|I|} < \frac{df_{i,J}}{|J|} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$fl(i,j) = \begin{cases} 1 & \text{if } \frac{df_{j,I}}{|I|} \geqslant \frac{df_{i,J}}{|J|} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We conducted an evaluation experiment using a set of 1995 artists from 9 genres extracted from *All Music Guide*. As ground truth we used the so-called "tiers" that reflect the importance, quality, and relevance of an artist to the respective genre, judged by *All Music Guide*'s editors, cf. amgabout (2007). Calculating *Spearman's rank-order correlation*, e.g., Sheskin (2004), between the ranking given by Formula (2) and the ranking given by *All Music Guide*'s tiers, revealed an average correlation coefficient of 0.38 over all genres. More details on the evaluation can be found in Schedl, Knees, and Widmer (2006).

To give an example of how the penalization term influences the ranking, we first consider the band "Tool", which is classified as "Heavy Metal" by *All Music Guide*'s editors.[5] This band has a backlink/forward link-ratio of $\frac{263}{8} = 32.875$ when applying Formula (2) without the $penalty(i)$ term. As a result, "Tool" ranks 3rd in the prototypicality ranking for the genre "Heavy Metal" (only superseded by "Death" and "Europe"), which we and also *All Music Guide*'s editors believe does not properly reflect the band's true importance for the genre, even though "Tool" is certainly no unknown band to the metal aficionado. However, when multiplying the ratio with the penalization term, which is 0.1578 for "Tool" (according to Formula (3)), the band is downranked to rank number 29 (of 271), which seems more accurate. In contrast, the artist "Alice Cooper", who obviously does not equal a common speech word, has a backlink/forward link-ratio of $\frac{247}{24} = 10.29$, which translates to rank 10. With a value of 0.8883 for Formula (3), "Alice Cooper" still remains at the 10th rank after applying the penalization factor, which we would judge highly accurate.

### 3.2. Band member and instrumentation detection

Another type of information indispensible for a music information system is *band members and instrumentation*. In order to capture such aspects, we first apply to the Web pages retrieved for a band a named entity detection (NED) approach. To this end, we extract all 2-, 3-, and 4-grams, assuming that the complete name of any band member does comprise of at least two and at most four single names. We then discard all $n$-grams whose tokens contain only one character and retain only the $n$-grams with their first letter in upper case and all other letters in lower case. Finally, we use the *iSpell English Word Lists* (isp, 2006) to filter out all $n$-grams where at least one token equals a common speech word. This last step in the NED is essential to suppress noise in the data, since in Web pages, word capitalization is used not only to denote named entities, but often also for highlighting purposes. The remaining $n$-grams are regarded as potential band members.

Subsequently, we perform shallow linguistic analysis to obtain the actual instrument(s) of each member. To this end, a set of seven patterns, like "`M`, the `R`" or "`M` plays the `I`", where $M$ is the potential member, $I$ is the instrument, and $R$ is the member's role in the band, is applied to the $n$-grams and the surrounding text as necessary. For $I$ and $R$, we use lists of synonyms to cope with the use of different terms for the same concept (e.g., "drummer" and "percussionist"). We then calculate the

[4] Terms like *Kiss*, *Bush*, or *Hole* often occur on (artist-related) Web pages, but do not necessarily denote the respective bands.

[5] In this example, we use the same data set of 1995 artists as in Schedl, Knees, and Widmer (2006).

document frequencies of the patterns and accumulate them over all seven patterns for each $(M,I)$-tuple. In order to suppress uncertain information, we filter out those $(M,I)$-pairs whose document frequency falls below a dynamic threshold $t_f$, which is parametrized by a constant $f$. $t_f$ is expressed as a fraction $f$ of the highest document frequency of any $(M,I)$-pair for the band under consideration. Consider, for example, a band whose top-ranked singer, according to the DF measure, has an accumulated DF count of 20. Using $f = 0.06$, all potential members with an aggregated DF of less than 2 would be filtered out in this case as $t_{0.06} = 20 \cdot 0.06 = 1.2$. The remaining tuples are predicted as members of the band under consideration. Note that this approach allows for an $m$:$n$ assignment between instruments and bands.

An evaluation of this approach was conducted on a data set of 51 bands with 499 members (current and former ones). The ground truth was gathered from *Wikipedia* (wik, 2009), *All Music Guide*, *discogs* (dis, 2009), or the band's Web site. We also assessed different query schemes to obtain *Google*'s top-ranked Web pages for each band:

- "`band`" + `music` (abbr. *M*)
- "`band`" + `music` + `review` (abbr. *MR*)
- "`band`" + `music` + `members` (abbr. *MM*)
- "`band`" + `music` + `lineup` (abbr. *LUM*)

Varying the parameter $f$, we can adjust the trade-off between precision and recall, which is depicted in Fig. 1. From the figure, we can see that the query schemes *M* and *MM* outperform the other two schemes. Another finding is that $f$ values in the range $[0.2, 0.25]$ (depending on query scheme) maximize the sum of precision and recall, at least for the used data set. Considering that there exists an upper limit for the recall achievable with our approach, due to the fact that usually not all band members are covered by the fetched 100 Web pages per artist, these results are pretty promising. The upper limit for the recall for the various query schemes is: *M*: 53%, *MR*: 47%, *MM*: 56%, *LUM*: 55%. For more details on the evaluation, a comprehensive discussion of the results, and a second evaluation taking only current band members into account, the interested reader is invited to consider Schedl and Widmer (2007).

### 3.3. Automatic tagging of artists

We perform automatically attributing textual descriptors to artists, commonly referred to as *tagging*, using a dictionary of about 1500 musically relevant terms in the indexing process. This dictionary resembles the one used in Pampalk et al. (2005). It contains terms somehow related to music, e.g., names of musical instruments, genres, styles, moods, time periods, and geographical locations. The dictionary is available at http://www.cp.jku.at/people/schedl/music/cob_terms.txt.

As for term selection, i.e., finding the most descriptive terms for an artist, we investigated three different term weighting measures (DF, TF, and TF·IDF) in a quantitative user study using a collection of 112 well-known artists (14 genres, 8 artists each), cf. Schedl and Pohle (2010). To this end, the 10 most important terms according to each term weighting function had been determined. In order to avoid biasing of the results, the 10 terms obtained by each weighting function were then merged into one list per artist. Hence, every participant was presented a list of 112 artist names and, for each name, the corresponding term list. Since the authors had no a priori knowledge of which artists were known by which participant, the participants were told to evaluate only those artists they were familiar with. Their task was then to rate the associated terms with respect to their appropriateness for describing the artist or his/her music. To this end, they had to associate every term to one of the three classes + (*good description*), − (*bad description*), and ∼ (indifferent or not wrong, but not a description specific for the artist). We had five participants in the user study and received a total of 172 assessments. Mapping the ratings in class + to the value 1, those in class − to −1, and those in class ∼ to 0 and calculating the arithmetic mean of the values of all

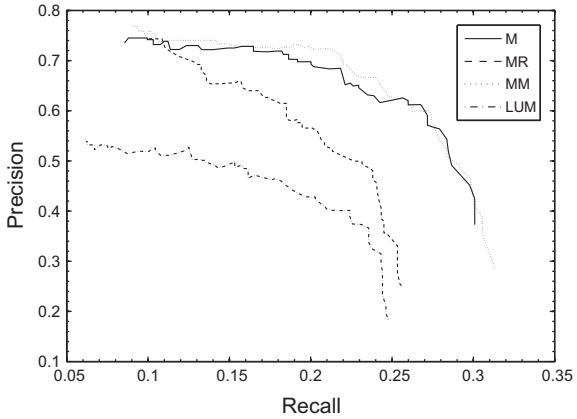


**Fig. 1.** Precision/recall-plot of the approach to band member and instrumentation detection.

**Table 1**
Results of Friedman's test to assess the significance of the differences in the term weighting measures.

| | |
|---|---|
| $N$ | 92 |
| $df$ | 2 |
| $\chi^2$ | 16.640 |
| $p$ | 0.00000236 |

assessments for each artist, we obtained a score representing the average excess of the number of good terms over the number of bad terms. These scores were 2.22, 2.43, and 1.53 for TF, DF, and TF·IDF, respectively.

To test for the significance of the results, we performed *Friedman's two-way analysis of variance* (Friedman & March, 1940; Sheskin, 2004). This test is similar to the two-way ANOVA, but does not assume a normal distribution of the data. It is hence a non-parametric test, and it requires related samples (ensured by the fact that for each artist all three measures were rated). The outcome of the test is summarized in Table 1. Due to the very low $p$ value, we can state that the variance differences in the results are significant with a very high probability. To assess which term weighting measures produce significantly different results, we conducted pairwise comparison between the results given by the three weighting functions. To this end, we employed the *Wilcoxon signed ranks test* (Wilcoxon, 1945) and tested for a significance level of 0.01. The test showed that TF·IDF performed significantly worse than both TF and DF, whereas no significant difference could be made out between the results obtained using DF and those obtained using TF. This result is quite surprising as TF·IDF is a well-established term weighting measure and commonly used to describe text documents according to the vector space model, cf. Salton, Wong, and Yang (1975). A possible explanation for the worse performance of TF·IDF is that this measure assigns high weights to terms that are very specific for a certain artist (high TF and low DF), which is obviously a desired property when it comes to distinguish one artist from another. In our application scenario, however, we aim at finding the most descriptive terms – not the most discriminative ones – for a given artist. This kind of terms seems to be better determined by the simple TF and DF measures. Hence, for the AGMIS application, we opted for the DF weighting to automatically select the most appropriate tags for each artist.

### 3.4. Co-Occurrence Browser

To easily access the top-ranked Web pages of any artist, we designed a user interface called *Co-Occurrence Browser* (COB), cf. Fig. 2. COB is based on the *Sunburst* visualization technique (Andrews & Heidegger, 1998; Stasko & Zhang, 2000), which we brought to the third dimension. The purpose of COB is threefold: First, it facilitates getting an overview of the set of Web pages related to an artist by structuring and visualizing them according to co-occurring terms. Second, it reveals meta-information about an artist through the descriptive terms extracted from the artist's Web pages. Third, by extracting the multimedia contents from the set of the artist's Web pages and displaying them via the COB, the user can explore the Web pages by means of audio, image, and video data.

In short, based on the dictionary used for automatic tagging, COB groups the Web pages of the artist under consideration with respect to co-occurring terms and ranks the resulting groups by their document frequencies.[6] The sets of Web pages are then visualized using the approach presented in Schedl, Knees, Widmer, Seyerlehner, and Pohle (2007). In this way, COB allows for browsing the artist's Web pages by means of descriptive terms. Information on the amount of multimedia content is encoded in the arcs' height, where each Sunburst visualization accounts for a specific kind of multimedia data. Thus, in Fig. 2, the topmost Sunburst represents the video content, the middle one the image content, and the lower one the audio content found on the respective Web pages.

### 3.5. Album cover retrieval

We presented preliminary attempts to automatically retrieve album cover artwork in Schedl, Knees, Pohle, and Widmer (2006). For the article at hand, we refined our approach and conducted experiments with content-based methods (using image processing techniques) as well as with context-based methods (using text mining) for detecting images of album covers on the retrieved Web pages. The best performing strategy, which we therefore employed to build AGMIS, uses the text distance between artist and album name and ⟨img⟩ tag as indicator for the respective image's likelihood of showing the sought album cover. To this end, we create a *word-level index* (Zobel & Moffat, 2006) that does not only contain the plain text, but also the HTML tags of the retrieved Web pages. After having filtered all images that are unlikely to show an album cover, as described below, we output the image with minimum distance between ⟨img⟩ tag and artist name and ⟨img⟩ tag and album name on the set of Web pages retrieved for the artist under consideration. Formally, the selection function is given in Formula (6), where $pos_i(t)$ denotes the offset of term $t$, i.e., its position $i$ in the Web page $p$, and $P_a$ denotes all pages retrieved for artist $a$.

[6] Any term weighting measure can be used, but the simple DF measure seemed to capture the most relevant terms best, cf. Section 3.3.

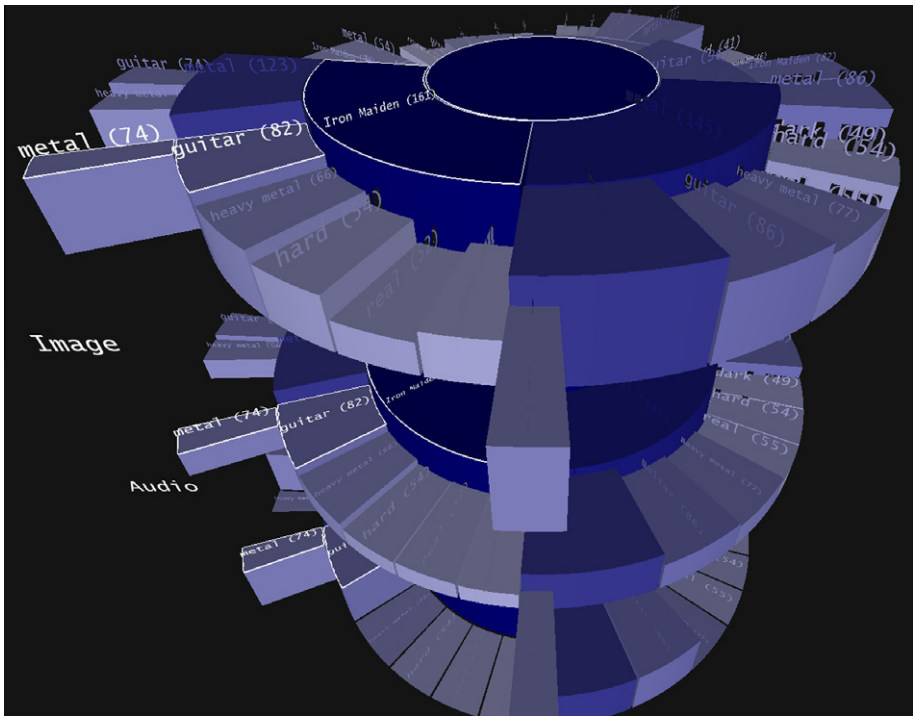


**Fig. 2.** COB visualizing a collection of Web pages retrieved for the band *Iron Maiden*.

$$min_{i,j,k}|pos_i(\langle img\rangle tag) - pos_j(artist\ name)| + |pos_i(\langle img\rangle tag) - pos_k(album\ name)| \quad \forall p \in P_a \tag{6}$$

As for filtering obviously erroneous images, content-based analysis is performed. Taking the almost quadratic shape of most album covers into account, all cover images that have non-quadratic dimensions within a tolerance of 15% are rejected. Since images of scanned compact discs often score highly on the text distance function, we use a circle detection technique to filter out those false positives. Usually, images of scanned discs are cropped to the circle-shaped border of the compact disc, which allows to use a simple circle detection algorithm. To this end, small rectangular regions along a circular path that is touched by the image borders tangentially are examined, and the contrast between subareas of these regions is determined using RGB color histograms. Since images of scanned compact discs show a strong contrast between subareas showing the imprint and subareas showing the background, the pixel distributions in the highest color value bins of the histograms are accumulated for either type of region (imprint and background). If the number of pixels in the accumulated imprint bins exceeds or falls short of the number of pixels in the accumulated background bins by more than a factor of 10, this gives strong evidence that the image under evaluation shows a scanned disc. In this case, the respective image is discarded.

On a test collection of 255 albums by 118 distinct, mostly European and American artists, our approach achieved a precision of up to 89% at a recall level of 93%, precision being defined as the number of correctly identified cover images among all predicted images, recall being defined as the number of found images among all albums in the collection. On a more challenging collection of 3311 albums by 1593 artists from all over the world, the approach yielded precision values of up to 73% at a recall level of 80%.

## 4. An automatically generated music information system

Since we aimed at building a music information system with broad artist coverage, we first had to gather a sufficiently large list of artists, on which the methods described in the previous section were applied. To this end, we extracted from *All Music Guide* nearly 700,000 music artists, organized in 18 different genres. In a subsequent data preprocessing step, all artists that were mapped to identical strings after *non-character removal*[7] were discarded, except for one occurrence. Table 2 lists the genre distribution of the remaining 636,475 artists according to *All Music Guide*, measured as absolute number of artists in each genre and as percentage in the complete collection. The notably high number of artists in the genre "Rock" can be explained by the large diversity of different music styles within this genre. In fact, taking a closer look at the artists subsumed in the genre "Rock" reveals pop artists as well as death metal bands. Nevertheless, gathering artist names from *All Music Guide* seemed the most reasonable solution to obtain a real-world artist list.

[7] This filtering was performed to cope with ambiguous spellings for the same artist, e.g., "B.B. King" and "BB King".

**Table 2**
List of genres used in AGMIS with the corresponding number of artists and their share in the complete collection as well as the number of artists for which no Web pages were found (*0-PC*).

| Genre | Artists | % | *0-PC* | % |
|---|---|---|---|---|
| Avantgarde | 4469 | 0.70 | 583 | 13.05 |
| Blues | 13,592 | 2.14 | 2003 | 14.74 |
| Celtic | 3861 | 0.61 | 464 | 12.02 |
| Classical | 11,285 | 1.77 | 1895 | 16.79 |
| Country | 16,307 | 2.56 | 2082 | 12.77 |
| Easy listening | 4987 | 0.78 | 865 | 17.35 |
| Electronica | 35,250 | 5.54 | 3101 | 8.80 |
| Folk | 13,757 | 2.16 | 2071 | 15.05 |
| Gospel | 26,436 | 4.15 | 5597 | 21.17 |
| Jazz | 63,621 | 10.00 | 10,866 | 17.08 |
| Latin | 33,797 | 5.31 | 9512 | 28.14 |
| New age | 13,347 | 2.10 | 2390 | 17.91 |
| Rap | 26,339 | 4.14 | 2773 | 10.53 |
| Reggae | 8552 | 1.34 | 1320 | 15.43 |
| RnB | 21,570 | 3.39 | 2817 | 13.06 |
| Rock | 267,845 | 42.08 | 39,431 | 14.72 |
| Vocal | 11,689 | 1.84 | 1988 | 17.01 |
| World | 59,771 | 9.39 | 17,513 | 29.30 |
| *Total* | 636,475 | 100.00 | 107,271 | 16.85 |

The sole input to the following data acquisition steps is the list of extracted artist names, except for the prototypicality estimation (cf. Section 3.1.2), which also requires genre information, and for the determination of album cover artwork (cf. Section 3.5), which requires album names. This additional information was also extracted from *All Music Guide*.

An overview of the data processing involved in building AGMIS is given in Fig. 3. The data acquisition process can be broadly divided into the three steps *querying* the search engine for the URLs of artist-related Web pages, *fetching* the HTML documents available at the retrieved URLs, and *indexing* the content of these documents.

*Querying*. We queried the *exalead* search engine for URLs of up to 100 top-ranked Web pages for each artist in the collection using the query scheme `"artist name" NEAR music`. The querying process took approximately one month. Its outcome was a list of 26,044,024 URLs that had to be fetched next.

*Fetching*. To fetch this large number of Web pages, we implemented a fetcher incorporating a load balancing algorithm to avoid excessive bandwidth consumption of servers frequently occurring in the URL list. The fetching process took approximately four and a half months. It yielded a total of 732.6 gigabytes of Web pages.

Some statistics concerning the retrieved Web pages give interesting insights. Table 2 shows, for each genre, the number of artists for which not a single Web page could be determined by the search engine, i.e., artists with a page count of zero. Not very surprisingly, the percentage is highest for the genres "Latin" and "World" (nearly 30% of zero-page-count-artists), which comprise many artists known only in regions of the world that are lacking broad availability of Internet access. In contrast, a lot of information seems to be available for artists in the genres "Electronica" and "Rap" (about 10% of 0-PC-artists). Table 3 depicts the number of Web pages retrieved for all artists per genre (column *RP*), the arithmetic mean of Web pages retrieved for an artist (column $RP_{\text{mean}}$), and the number of retrieved pages with a length of zero, i.e., pages that were empty or could not be fetched for some reason. Since the main reason for the occurrence of such pages were server errors, their relative frequencies are largely genre-independent, as it can be seen in the fifth column of Table 3. The table further shows the median and arithmetic mean of the page counts returned by *exalead* for the artists in each genre. These values give strong indication that artists in the genres "Latin", "Gospel", and "World" tend to be underrepresented on the Web.

*Indexing*. To create a *word-level index* of the retrieved Web pages (Zobel & Moffat, 2006), the open source indexer *Lucene Java* (luc, 2008) was taken as a basis and adapted by the authors to suit the HTML format of the input documents and the requirements for efficiently extracting the desired artist-related pieces of information.

Although indexing seems to be a straightforward task at first glance, we had to resolve certain issues. Foremost some heavily erroneous HTML files were encountered, which caused *Lucene* to hang or crash, and thus required special handling. More precisely, some HTML pages showed a size of tens of megabytes, but were largely filled with escape characters. To resolve these problems, a size limit of 5 megabytes for the HTML files to index was introduced. Additionally, a 255-byte-limit for the length of each token was used.

AGMIS makes use of two indexes. Creating the first one was performed applying neither stopping, nor stemming, nor casefolding as it is used for band member and instrumentation detection (cf. Section 3.2) and to calculate artist similarities (cf. Section 3.1.1). Since the patterns applied in the linguistic analysis step of our approach to band member detection contain a lot of stop words, applying stopping either would have been virtually useless (when using a stop word list whose entries were corrected for the words appearing in the patterns) or would have yielded a loss of information crucial to the application of the patterns. Since artist names sought for in our approach to similarity estimation typically also contain stop words, applying stopping would be counterproductive for this purpose as well. The size of the optimized, compressed first index is 228 gigabytes. A second index containing only the terms in the music dictionary was created to generate term

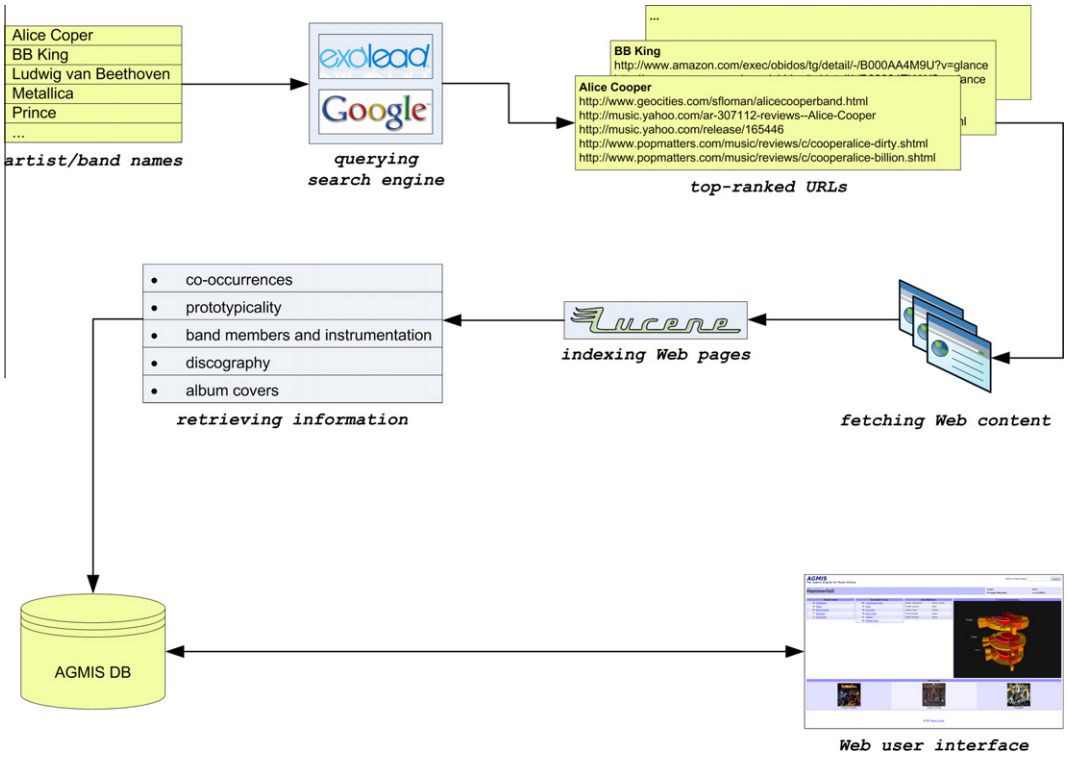


**Fig. 3.** Data processing diagram of AGMIS.

**Table 3**
The number of retrieved Web pages per genre (*RP*) and its mean per artist ($RP_{mean}$), the number of empty Web pages among them (*0-L*), and the median and mean of available Web pages according to the page-count-value returned by the search engine ($PC_{med}$ and $PC_{mean}$).

| Genre | RP | $RP_{mean}$ | *0-L* | % | $PC_{med}$ | $PC_{mean}$ |
|---|---|---|---|---|---|---|
| Avantgarde | 204,870 | 46 | 32,704 | 15.96 | 29 | 14,969 |
| Blues | 554,084 | 40 | 89,832 | 16.21 | 18 | 2893 |
| Celtic | 136,244 | 35 | 23,627 | 17.34 | 25 | 5415 |
| Classical | 509,269 | 45 | 99,181 | 19.48 | 27 | 4149 |
| Country | 696,791 | 42 | 116,299 | 16.69 | 22 | 2562 |
| Easy listening | 187,749 | 37 | 32,758 | 17.45 | 14 | 4808 |
| Electronica | 1,973,601 | 56 | 317,863 | 16.11 | 65 | 31,366 |
| Folk | 544,687 | 39 | 89,385 | 16.41 | 18 | 5166 |
| Gospel | 876,017 | 33 | 142,690 | 16.29 | 8 | 4791 |
| Jazz | 2,306,785 | 36 | 361,160 | 15.66 | 13 | 6720 |
| Latin | 866,492 | 25 | 139,660 | 16.12 | 4 | 19,384 |
| New age | 488,799 | 36 | 82,075 | 16.79 | 13 | 12,343 |
| Rap | 1,322,187 | 50 | 223,052 | 16.87 | 37 | 38,002 |
| Reggae | 377,355 | 44 | 58,180 | 15.42 | 22 | 16,000 |
| RnB | 898,787 | 41 | 141,339 | 15.73 | 17 | 17,361 |
| Rock | 12,058,028 | 43 | 1,908,904 | 15.83 | 21 | 16,085 |
| Vocal | 461,374 | 39 | 77,073 | 16.71 | 15 | 10,421 |
| World | 1,577,769 | 26 | 257,649 | 16.33 | 4 | 14,753 |
| *Total* | 26,040,888 | 40 | 4,193,431 | 16.10 | 16 | 15,120 |

profiles for the purpose of artist tagging (cf. Section 3.3) and for the COB (cf. Section 3.4). The size of this index amounts to 28 gigabytes.

### *4.1. AGMIS' user interface*

The pieces of information extracted from the artist-related Web pages and inserted into a relational *MySQL* (mys, 2008) database are offered to the user of AGMIS via a Web service built on *Java Servlet* and *Java Applet* technology. The home page of the AGMIS Web site reflects a quite simple design, like the one used by *Google*. Besides a brief explanation of the system, it

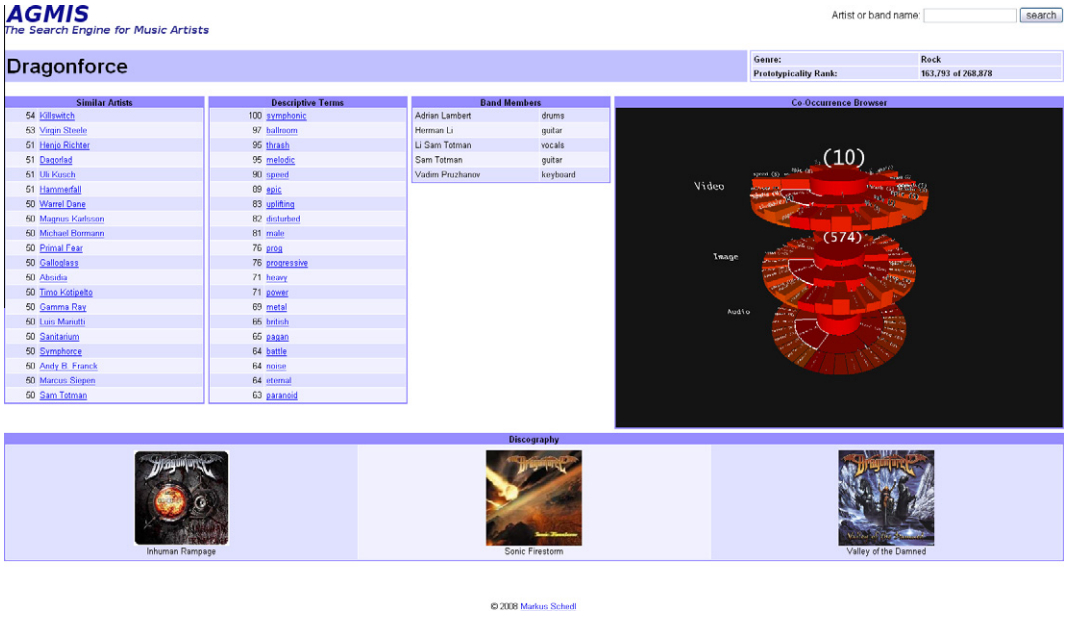


**Fig. 4.** The user interface provided by AGMIS for the band *Dragonforce*.

only displays a search form, where the user can enter an artist or band name. To allow for fuzzy search, the string entered by the user is compared to the respective database entries using *Jaro-Winkler similarity*, cf. (Cohen, Ravikumar, & Fienberg, 2003). The user is then provided a list of approximately matching artist names, from which he or she can select one.

After the user has selected the desired artist, AGMIS delivers an artist information page. Fig. 4 shows an example of such a page for the band *Dragonforce*. On the top of the page, artist name, genre, and prototypicality rank are shown. Below this header, lists of similar artists, of descriptive terms, and of band members and instrumentation, where available and applicable, are shown. As a matter of course, the information pages of similar artists are made available via hyperlinks. Moreover, it is also possible to search for artists via descriptive terms. By clicking on the desired term, AGMIS starts searching for artists that have this term within their set of highest ranked terms and subsequently displays a selection list. To the right of the lists described so far, the Co-Occurrence Browser is integrated into the user interface as a *Java Applet* to permit browsing the indexed Web pages and their multimedia content. The lower part of the artist information page is dedicated to discography information, i.e., a list of album names and album cover images are shown.

### 4.2. Computational complexity

Most tasks necessary to build AGMIS were quite time-consuming. The querying, fetching, and indexing processes, the creation of artist term profiles, the calculation of term weights, and all information extraction tasks were performed on two standard personal computers with *Pentium 4* processors clocked at 3 GHz, 2 GB RAM, and a RAID-5 storage array providing 2 TB of usable space. In addition, a considerable amount of external hard disks serving as temporary storage facilities were required.

#### 4.2.1. Running times

In Table 4, precise running times for indexing, information extraction, and database operation tasks are shown for those tasks for which we measured the time. Calculating the artist similarity matrix was carried out as follows. Computing the complete 636,475 $\times$ 636,475 similarity matrix requires 202,549,894,575 pairwise similarity calculations. Although performing this number of calculations is feasible in reasonable time on a current personal computer in regard to computational power, the challenge is to have the required vectors in memory when they are needed. As the size of the complete similarity matrix amounts to nearly 800 gigabytes, even when storing symmetric elements only once, it is not possible to hold all data in memory. Therefore, we first split the 636,475 $\times$ 636,475 matrix into 50 rows and 50 columns, yielding 1275 submatrices when storing symmetric elements only once. Each submatrix requires 622 megabytes and thus fits well into memory. Artist similarities were then calculated between the 12,730 artists in each submatrix, processing one submatrix at a time. Aggregating these submatrices, individual artist similarity vectors were extracted, and the most similar artists for each artist in the collection were selected and inserted into the database.

#### 4.2.2. Asymptotic runtime complexity

The asymptotic runtime complexities of the methods presented in Section 3 are summarized in Table 5, supposing that querying, fetching, and indexing was already performed. Querying is obviously linear (in terms of issued requests) in the

**Table 4**
Some running times of tasks performed while creating AGMIS.

| Task | Time (s) |
|---|---|
| Creating *Lucene* index using all terms (no stopping, stemming, casefolding) | 218,681 |
| Creating *Lucene* index using the music dictionary | 211,354 |
| Computing the term weights (TF, DF, and TF·IDF) | 514,157 |
| Sorting the terms for each artist and each weighting function | 13,503 |
| Computing the artist similarity matrix via submatrices | 2,489,576 |
| Extracting artist similarity vectors from the submatrices | 3,011,719 |
| Estimating artist prototypicalities by querying *exalead* | 4,177,822 |
| Retrieving album cover artwork | 6,654,703 |
| Retrieving information on multimedia content (audio, image, video) for the COB | 2,627,369 |
| Retrieving band members and instrumentation for artists in genre "Rock" | 213,570 |
| Importing the 20 most similar artists for each artist into the AGMIS database | 356,195 |
| Importing the 20 top-ranked terms for each artist into the AGMIS database | 3649 |
| Importing album names and covers into the AGMIS database | 6686 |

**Table 5**
Asymptotic runtime complexities of the IE approaches.

| Task | Runtime complexity |
|---|---|
| Artist similarity calculation | $\mathcal{O}(n^2 \cdot \log k)$ |
| Artist prototypicality estimation | $\mathcal{O}(n^2 \cdot \log k)$ |
| Band member and instumentation detection | $\mathcal{O}(n \cdot k \cdot p)$ |
| Artist tagging | $\mathcal{O}(n \cdot k)$ |
| Album cover retrieval | $\mathcal{O}(n \cdot k)$ |

number of artists, i.e., $\mathcal{O}(n)$, provided that the desired number of top-ranked search results $p$ retrieved per artist does not exceed the number of results that can be returned by the search engine in one page. Fetching can be performed in $\mathcal{O}(n \cdot p)$, but will usually require less operations (cf. Table 2, the average number of Web pages retrieved per artist is 40). Using a *B-tree* (Bayer, 1971) as data structure, indexing can be performed in $\mathcal{O}(t \cdot \log t)$, where $t$ is the total number of terms to be processed.

In Table 5, $n$ denotes the total number of artists and $k$ the total number of keys in the index. Creating the symmetric similarity matrix and estimating the prototypicality for each artist both require $n^2$ requests to the index. Since each request takes $\log k$, the complexity of the whole process is $\mathcal{O}(n^2 \cdot \log k)$. The band member detection requires $k$ operations to extract the potential band members, i.e., $n$-grams, for each of which $p$ operations are needed to evaluate the patterns and obtain their document frequencies, $p$ being the number of patterns in all variations, i.e., all synonyms for instruments and roles counted as a separate pattern (cf. Section 3.2). In total, the asymptotic runtime complexity is therefore $\mathcal{O}(n \cdot k \cdot p)$. The automatic artist tagging procedure is in $\mathcal{O}(n \cdot k)$, where $k$ is again the number of terms in the index. However, as we use a dedicated index for the purpose of artist tagging, $k \approx 1500$, and therefore $k \ll n$. Finally, the current implementation of our album cover retrieval technique requires $n \cdot k$ operations, since all keys in the index have to be sought for ⟨img⟩ tags, artist names, and album names. This could be sped up by building an optimized index with clustered ⟨img⟩ tags, which will be part of future work.

## 5. Conclusions and future work

This article has given an overview of state-of-the-art techniques for Web-based information extraction in the music domain. In particular, techniques to mine relations between artists (similarities and prototypicality), band members and instrumentation, descriptive terms, and album covers were presented. Furthermore, this article briefly described the *Co-Occurrence Browser* (COB), a user interface to organize and access artist-related Web pages via important, music-related terms and multimedia content. It was further shown that the proposed approaches can be successfully applied on a large scale using a real-world database of more than 600,000 music artists. Integrating the extracted information into a single information system yielded the *Automatically Generated Music Information System* (AGMIS), whose purpose is to provide access to the large amount of data gathered. The design, implementation, and feeding of the system were reported in detail.

Even though the evaluation experiments conducted to assess the techniques underlying AGMIS showed promising results, they still leave room for improvement in various directions. First, Web page retrieval could be pursued using focused crawling instead of directed search via search engines. This would presumably yield more accurate results, while at the same time limit Web traffic. Second, deep natural language processing techniques and more sophisticated approaches to named entity detection and machine learning could be employed to derive more specific information, especially in band member and instrumentation detection as well as to obtain detailed discography information. For example, temporal information would allow for creating band and artist histories as well as time-dependent relationship networks. Automatically generated biographies would be the ultimate aim. Finally, the information gathered by the Web mining techniques presented here could be

complemented with information extracted from the audio signal. Audio signal-based similarity information at the track level would enable enhanced services and applications, like automatic playlist generation or user interfaces to explore huge music collections in virtual spaces. Bringing AGMIS to the track level would also permit to provide song lyrics since approaches to automatically extracting a correct version of a song's lyrics do already exist, cf. Korst and Geleijnse (2006) and Knees et al. (2005). Employing methods to align audio and lyrics could eventually even allow for applications like an automatic karaoke system.

## Acknowledgments

This research is supported by the *Austrian Science Fund (FWF)* under Project Numbers L511-N15, Z159, and P22856-N23. The authors would further like to thank *Julien Carcenac* from *exalead* for his support in the querying process.

## References

Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., et al (2003). Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems, 18*(1), 14–21.

amgabout (2007). <http://www.allmusic.com/cg/amg.dll?p=amg&sql=32:amg/info_pages/a_about.html> Accessed November 2007.

amg (2009). <http://www.allmusic.com> Accessed November 2009.

Andrews, K., & Heidegger, H. (1998). Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proceedings of IEEE information visualization 1998*. Research Triangle Park, NC, USA.

Aucouturier, J.-J., & Pachet, F. (2002). Scaling up music playlist generation. In *Proceedings of the IEEE international conference on multimedia and expo (ICME 2002)* (pp. 105–108). Lausanne, Switzerland.

Aucouturier, J.-J., & Pachet, F. (2004). Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences, 1*(1).

Aucouturier, J.-J., Pachet, F., & Sandler, M. (2005). The way it sounds: Timbre models for analysis and retrieval of music signals. *IEEE Transactions on Multimedia, 7*(6), 1028–1035.

Bayer, R. (1971). Binary B-trees for virtual memory. In *Proceedings of the ACM SIG FIDET workshop*. San Diego, CA, USA.

Cano, P., & Koppenberger, M. (2004). The emergence of complex network patterns in music artist networks. In *Proceedings of the 5th international symposium on music information retrieval (ISMIR 2004)* (pp. 466–469). Barcelona, Spain.

Celma, O. (2008). Music recommendation and discovery in the long tail. Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain. <http://mtg.upf.edu/ocelma/PhD/doc/ocelma-thesis.pdf>.

Celma, O., & Lamere, P. (2007). ISMIR 2007 tutorial: Music recommendation. <http://mtg.upf.edu/~ocelma/MusicRecommendationTutorial-ISMIR2007> Accessed December 2007.

Chakrabarti, S., van den Berg, M., & Dom, B. (1999). Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks, 31*(11–16), 1623–1640.

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-03 workshop on information integration on the web (IIWeb-03)* (pp. 73–78). Acapulco, Mexico.

Cohen, W. W., & Fan, W. (2000). Web-collaborative filtering: Recommending music by crawling the web. *WWW9/Computer Networks, 33*(1–6), 685–698.

dis (2009). <http://www.discogs.com> Accessed October 2009.

Dixon, S., Gouyon, F., & Widmer, G. (2004). Towards characterisation of music via rhythmic patterns. In *Proceedings of the 5th international symposium on music information retrieval (ISMIR 2004)* (pp. 509–516). Barcelona, Spain.

Downie, J. S. (2003). Music information retrieval. *Annual Review of Information Science and Technology, 37*, 295–340.

Eck, D., Bertin-Mahieux, T., & Lamere, P. (2007). Autotagging music using supervised machine learning. In *Proceedings of the 8th international conference on music information retrieval (ISMIR 2007)*. Vienna, Austria.

Ellis, D. P., Whitman, B., Berenzweig, A., & Lawrence, S. (2002). The quest for ground truth in musical artist similarity. In *Proceedings of 3rd international conference on music information retrieval (ISMIR 2002)*. Paris, France.

exa (2009). <http://www.exalead.com> Accessed February 2009.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92.

Geleijnse, G., & Korst, J. (2006). Web-based artist categorization. In *Proceedings of the 7th international conference on music information retrieval (ISMIR 2006)*. Victoria, Canada.

Geleijnse, G., Schedl, M., & Knees, P. (2007). The quest for ground truth in musical artist tagging in the social web era. In *Proceedings of the 8th international conference on music information retrieval (ISMIR 2007)*. Vienna, Austria.

goo (2009). <http://www.google.com> Accessed March 2009.

Hu, X., Bay, M., & Downie, J. S. (2007). Creating a simplified music mood classification ground-truth set. In *Proceedings of the 8th international conference on music information retrieval (ISMIR 2007)*. Vienna, Austria.

isp (2006). <http://wordlist.sourceforge.net> Accessed June 2006.

Knees, P., Pampalk, E., & Widmer, G. (2004). Artist classification with web-based data. In *Proceedings of the 5th international symposium on music information retrieval (ISMIR 2004)* (pp. 517–524). Barcelona, Spain.

Knees, P., Schedl, M., & Widmer, G. (2005). Multiple lyrics alignment: automatic retrieval of song lyrics. In *Proceedings of 6th international conference on music information retrieval (ISMIR 2005)* (pp. 564–569). London, UK.

Knees, P., Schedl, M., Pohle, T., & Widmer, G. (2006). 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.

Knees, P., Pohle, T., Schedl, M., & 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 2007)*. Amsterdam, the Netherlands.

Knees, P., Schedl, M., Pohle, T., & Widmer, G. (2007). Exploring music collections in virtual landscapes. *IEEE MultiMedia, 14*(3), 46–54.

Korst, J., & Geleijnse, G. (2006). Efficient lyrics retrieval and alignment. In W. Verhaegh, E. Aarts, W. ten Kate, J. Korst, & S. Pauws (Eds.), *Proceedings of the 3rd Philips symposium on intelligent algorithms (SOIA 2006)* (pp. 205–218). Eindhoven, the Netherlands.

Lamere, P. (2008). Social tagging and music information retrieval. *Journal of New Music Research: Special Issue: From Genres to Tags: Music Information Retrieval in the Age of Social Tagging, 37*(2), 101–114.

las (2009). <http://last.fm> Accessed February 2009.

Law, E., von Ahn, L., Dannenberg, R., & Crawford, M. (2007). Tagatune: A game for music and sound annotation. In *Proceedings of the 8th international conference on music information retrieval (ISMIR 2007)*. Vienna, Austria.

luc (2008). <http://lucene.apache.org> Accessed January 2008.

Mandel, M. I., & Ellis, D. P. (2005). Song-level features and support vector machines for music classification. In *Proceedings of the 6th international conference on music information retrieval (ISMIR 2005)*. London, UK.

Mandel, M. I., & Ellis, D. P. (2007). A web-based game for collecting music metadata. In *Proceedings of the 8th international conference on music information retrieval (ISMIR 2007)*. Vienna, Austria.
mys (2008). <http://www.mysql.com> Accessed June 2008.
Pachet, F., Westerman, G., & Laigre, D. (2001). Musical data mining for electronic music distribution. In *Proceedings of the 1st international conference on web delivering of music (WEDELMUSIC 2001)*. Florence, Italy.
Pampalk, E., & Goto, M. (2007). MusicSun: A new approach to artist recommendation. In *Proceedings of the 8th international conference on music information retrieval (ISMIR 2007)*. Vienna, Austria.
Pampalk, E., Rauber, A., & Merkl, D. (2002). Content-based organization and visualization of music archives. In *Proceedings of the 10th ACM international conference on multimedia (MM 2002)* (pp. 570–579). Juan les Pins, France.
Pampalk, E., Flexer, A., & 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 2005)*. Vienna, Austria.
Pohle, T., Knees, P., Schedl, M., Pampalk, E., & Widmer, G. (2007). Reinventing the wheel: A novel approach to music player interfaces. *IEEE Transactions on Multimedia, 9*, 567–575.
Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.
Schedl, M., & Pohle, T. (2010). Enlightening the sun: A user interface to explore music artists via multimedia content. *Multimedia Tools and Applications: Special Issue on Semantic and Digital Media Technologies, 49*(1), 101–118.
Schedl, M., & 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 2007)*. Paris, France.
Schedl, M., Knees, P., & Widmer, G. (2005a). 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.
Schedl, M., Knees, P., & Widmer, G. (2005b). Discovering and visualizing prototypical artists by web-based co-occurrence analysis. In: Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005). London, UK.
Schedl, M., Knees, P., Pohle, T., & Widmer, G. (2006). Towards automatic retrieval of album covers. In *Proceedings of the 28th European conference on information retrieval (ECIR 2006)*. London, UK.
Schedl, M., Knees, P., & Widmer, G. (2006). Investigating web-based approaches to revealing prototypical music artists in genre taxonomies. In *Proceedings of the 1st IEEE international conference on digital information management (ICDIM 2006)*. Bangalore, India.
Schedl, M., Pohle, T., Knees, P., & Widmer, G. (2006c). Assigning and visualizing music genres by web-based co-occurrence analysis. In *Proceedings of the 7th international conference on music information retrieval (ISMIR 2006)*. Victoria, Canada.
Schedl, M., Knees, P., Widmer, G., Seyerlehner, K., & Pohle, T. (2007). Browsing the web using stacked three-dimensional sunbursts to visualize term co-occurrences and multimedia content. In *Proceedings of the 18th IEEE visualization 2007 conference (Vis 2007)*. Sacramento, CA, USA.
Sheskin, D. J. (2004). *Handbook of parametric and nonparametric statistical procedures* (3rd ed.). Boca Raton, London, New York, Washington, DC: Chapman & Hall/CRC.
Stasko, J., & Zhang, E. (2000). Focus + context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of IEEE information visualization 2000*. Salt Lake City, UT, USA.
Turnbull, D., Liu, R., Barrington, L., & 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 2007)*. Vienna, Austria.
von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *CHI'04: Proceedings of the SIGCHI conference on human factors in computing systems*. New York, NY, USA: ACM Press.
Whitman, B., & Lawrence, S. (2002). 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.
wik (2009). <http://www.wikipedia.org> Accessed December 2009.
Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), 80–83.
Zadel, M., & Fujinaga, I. (2004). Web services for music information retrieval. In *Proceedings of the 5th international symposium on music information retrieval (ISMIR 2004)*. Barcelona, Spain.
Zobel, J., & Moffat, A. (2006). Inverted files for text search engines. *ACM Computing Surveys, 38*, 1–56.

**Markus Schedl** holds a Ph.D. in computer science (computational perception) from the *Johannes Kepler University Linz*, where he is employed as assistant professor. He graduated in computer science from the *Vienna University of Technology*. His main research interests include Web mining, multimedia information retrieval, information visualization, and intelligent user interfaces.

**Gerhard Widmer** is professor and head of the *Department of Computational Perception* at the *Johannes Kepler University Linz* and head of the *Intelligent Music Processing and Machine Learning* group at the *Austrian Research Institute for Artificial Intelligence*, Vienna, Austria. He holds M.Sc. degrees from the *Vienna University of Technology* and the *University of Wisconsin*, Madison, and a Ph.D. in computer science from the *Vienna University of Technology*. His research interests are in machine learning, pattern recognition, and intelligent music processing. In 2009, he was awarded Austria's highest research prize, the Wittgenstein Prize, for his work on AI and music.

**Peter Knees** graduated in computer science. Since February 2005 he has been working as a project assistant at the *Johannes Kepler University Linz*. He performs research towards a doctoral thesis with a focus on music information retrieval. Since 2004 he has been studying psychology at the *University of Vienna*.

**Tim Pohle** holds a Dipl.-Inf. degree (equivalent to M.Sc. in computer science) from the *Technical University Kaiserslautern* and a Ph.D. in computer science from the *Johannes Kepler University Linz*. His main field of interest is music information retrieval with a special emphasis on audio based-techniques.