Web-Based Artist Categorization

Gijs Geleijnse **Jan Korst**
Philips Research
Prof Holstlaan 4
5656 AA Eindhoven (the Netherlands)
{gijs.geleijnse,jan.korst}@philips.com

Abstract

We present a novel approach in categorizing artists into subjective categories such as genre. We base our method on co-occurrences on the web, found with the Google search engine. A direct mapping between artists and categories proved to be unreliable. We use the categories mapped to closely related artists to obtain a more reliable mapping. The method is tested on a genre classification test set with convincing results. Moreover, mood categorization is explored using the same techniques.

Keywords: Artist categorization, web, Google.

1. Introduction

Web services in the music domain provide all kinds of meta-data on music. Some meta-data is objective and verifiable, like the year of release of an album or the nationality of an artist. Other meta-data concerns the categorization of music, such as the assignment of a genre to an album or a mood to a song. Although such information may be debatable, it can be helpful for a user in selecting music.

The World Wide Web provides an overwhelming amount of information on the music domain. Home pages of artists, fan pages and album reviews give information to identify categories (genres, moods) for music. Even if some of the sources contradict each other, we are often able to select the most appropriate category.

In this paper, we present a method to automatically categorize artists and their music into moods and genres. The method is based on co-occurrences on the eb. We are interested in finding the most appropriate mapping from a given set of categories.

In earlier work on artist classification with web data the number of Google hits for queries with two artists was used [12, 15]. In this work we compare such an approach with more efficient methods with respect to the number of Google queries. Contrary to approaches in [8, 12], the methods introduced here are simple and unsupervised.

This paper is organized as follows. In the next section the problem statement is given. Section 3 handles three alternative methods in acquiring a mapping between artists and categories. Since such a mapping showed to be unreliable, we acquire additional information and use this in a final mapping in Section 4. Two experiments are discussed in Section 5: the categorization of artists into genres and into moods. Related work can be found in Section 6 and we conclude in Section 7.

2. Problem Description

Given are two sets A and L. The set A consists of names of artists, and the set L contains categories e.g. genres. Also a mapping $m : A \rightarrow L$ is given. We assume the set L to be complete and consequently we assume that each u in A can be mapped to at least one v in L.

Definition. We call an element $m(u) \in L$ *the most appropriate* category for $u \in A$ if a domain expert would select $m(u)$ from the set L as the category best applicable to u.

Problem. Given a set of artists A and a set of categories L, find for each $u \in A$ the most appropriate mapping $m(u) \in L$.

We use the web to extract such information and assume that an artist categorization can be deduced from it.

3. Three Methods for Categorizing Artists

In this section, we present three methods to categorize artists using web data. The first method is based on analyzing the numbers of co-occurrences of artists in A and categories in L on the web. To retrieve this data we use the Google search engine [4].

The estimated numbers of Google hits can fluctuate which may lead to unexpected results [14]. Another drawback of this method is, that it requires many queries to a search engine. For large sets of instances this can be problematic, since search engines currently allow only a limited amount of automated queries per day.

In Sections 3.2 and 3.3 we present two alternative methods that do not suffer from these drawbacks.

3.1. Page-count-based mapping (PCM)

We are interested in a preliminary mapping m'. To obtain such a mapping we perform a Google query `"`a`","`g`"` for

each pair $(a, g) \in A \times L$. Per query, we extract the estimated number of *hits* $\text{co}(a, b)$.

$$\text{co}(a, b) = \text{‘the number of hits for query ”}a\text{”, ”}b\text{” ’}$$

We assume that the order of a and b in the query does not effect the number of hits, thus $\text{co}(a, b)$ equals $\text{co}(b, a)$.

This Page-Count-based Mapping (PCM) is simple and intuitive. If we are for example interested in categorizing music artist into genres, we analyze the number of hits to queries for combinations of the names of the artist and each genre. Assuming Johnny Cash to be a country artist, we expect that more documents contain both the terms *Country* and *Johnny Cash* than *Reggae* and *Johnny Cash*.

Per $a \in A$ we could map the $g \in L$ with the most hits. However, we observe that frequently occurring categories in L have a larger probability to be mapped to any artist in A. For example, the query ‘*Pop*’ results in 8 times more hits than the query ’*Disco*’. Although we consider *Boney M* as a disco-act, the query *Boney M, pop* gives twice the amount of hits as *Boney M, disco*. This observation leads to a normalized approach, inspired by the pointwise mutual information theory [9].

For $a \in A$ and $g \in L$, we define a scoring function $S(a, g)$ between the two as follows.

$$S(a, g) = \frac{\text{co}(a, g)}{1 + \sum_{b \in A} \text{co}(b, g)} \tag{1}$$

In the denominator we add 1 to the sum of all co-occurrences with g to avoid dividing by 0.

Now we have computed the scores for all pairs, we select a preliminary mapping m' for all $a \in A$. Per artist we select the category $g \in L$ with the highest score S.

$$m'(a) = \text{argmax}_{h \in L} S(a, h) \tag{2}$$

Using PCM we thus need to perform $|A| \cdot |L|$ queries.

3.2. Pattern-based mapping (PM)

This mapping (PM) is based on occurrences of terms in phrases on the web. We observe combinations of terms in phrases that express the relation we are interested in. For example, if we are interested in the relation between artists and their genres, an appropriate phrase that links terms of the two could be *‘(artist) is one of the biggest (genre) artists’*.

The method works as follows. We compose a number of patterns that express the categorization of the artists in A into categories of L. We can either compose these patterns manually, or use a training set to automatically find patterns [7].

We combine the patterns with an instance of either one of the arguments of the relation it reflects. We use these combinations of a pattern and an instance as a query to the search engine [7]. For example, if we have the pattern *“(genre) artist such as (artist)”*, we use it in queries in combinations with all names of genres and artists. For example, we use this pattern both for the query *“Country artists such as”* and for the query *“artists such as Prince”*. In the results for the first query, we identify artists in A, while in the results for the second query we search for genres in L related to *Prince*.

These queries provide access to relevant data. From the excerpts returned by the search engine, we thus identify the elements of either A or L to measure the number of co-occurrences of the pairs. Hence, using PM $\text{co}(a, b)$ is defined as follows.

$$\begin{aligned} \text{co}(a, b) = \quad & \text{‘number of occurrences of } a \\ & \text{when querying patterns containing } b\text{’} \; + \\ & \text{‘number of occurrences of } b \\ & \text{when querying patterns containing } a\text{’} \end{aligned}$$

Using PM we only need $m \cdot (|A| + |L|)$ queries, with m the number of patterns expressing the mapping m. We use the same scoring function (1) to obtain a preliminary mapping (2).

3.3. Document-based mapping (DM)

In the Document-based Mapping (DM) we collect the first k URLs of the documents returned by the search engine for some query. These k URLs are the most relevant for the query submitted based on the ranking used by the search engine [4].

In the first phase of the algorithm, we query all elements in both A and L and collect the top k documents for each of the queries. For the artists in A, we retrieve each document using the URLs found by the search engine. We count the occurrences of the categories in L in the retrieved documents for the intermediate mapping m'. From the documents retrieved with an instance $g \in L$, we similarly extract the occurrences of artists in A.

The documents obtained using DM are the most relevant for each element $a \in A$. For the artists queried we expect fan pages, the home page of the artist, entries in music database sites and so on. The genres or styles mentioned in these pages will most probably reflect the artist queried.

The co-occurrences function is here thus defined as follows.

$$\begin{aligned} \text{co}(a, b) = \quad & \text{‘number of occurrences of } a \\ & \text{in documents found with “}b\text{” ’} \; + \\ & \text{‘number of occurrences of } b \\ & \text{in documents found with “}a\text{” ’} \end{aligned}$$

The co-occurrences between artists and genres again are used for an intermediate mapping using the same scoring function.

This method requires only $|A| + |L|$ queries. However, additional data communication is required since each of the documents has to be downloaded instead of using only the data provided by the search engine.

4. Improving Precision using Additional Information

Since the preliminary mapping showed to be unreliable (see Section 5), we observe the need for additional data. We use the assumption that related artists often share the same category. The other way around, if two artists are both known for the same category (e.g. romantic music), we expect them to occur often in the same context. We thus use the working hypothesis that strongly related artists in general are categorized with the same element in L.

We acquire co-occurrences between artists (Section 4.1) using PCM, PM or DM as described in the previous section. We use this information in a final mapping m (Section 4.2).

4.1. Finding co-occurences between artists

We consider artists to be related, when they frequently occur in the same context. The methods PCM, PM and DM can be used to gather numbers of co-occurrences $\text{co}(a, b)$ between artists a and b.

Per pair $(a, b) \in A \times A$ we compute the score T, similar to the score S in (1).

$$T(a,b) = \frac{\text{co}(a,b)}{1 + \sum_{y,y\neq a} \text{co}(a,y) \cdot \sum_{x,x\neq b} \text{co}(x,b)} \qquad (3)$$

The scoring function T is symmetric in its arguments. Again, we do not use a majority voting to prevent frequently occurring instances to be strongly related to many other instances. For example, an artist like *Madonna* is expected to co-occur a lot with many other artists, due to large number of web pages mentioning *Madonna* (26 million hits).

In PCM we combine the names of two artists into a query and extract the number of *hits*. Using this method this phase requires $|A|^2$ queries. The total number of Google queries for the PCM method is thus $|A| \cdot (|A| + |L|)$.

If we use PM to obtain the numbers of co-occurrences between artists, we can specify the relatedness between artists. For example, we can be solely interested in artists who have played together. A pattern such as *"(artist) recorded a duet with (artist)"* could be suitable for this purpose. This phase of the method consists of $k \cdot |A|$ queries (with k the number of patterns), leading to a total Google complexity of $|A|+|L|$

In the documents obtained with the DM method we only expect names of other artists that are strongly connected with the artist queried. We reuse the documents obtained by querying the artists in the first phase. This method thus requires $|A| + |L|$ queries in total.

4.2. Combine results in final mapping

We combine the scores T with the preliminary mapping m' as follows. Per artist a, we inspect m' to determine the category that is assigned most often to a and its n closest related artists. We thus expect that the most appropriate category v for a is most often mapped by m' among the nearest neighbors to a.

Per instance $a \in A$, we construct an ordered list with a and its n nearest neighbors, $\mathcal{A} = (a_0, a_1, ..., a_n)$ with $a = a_0$ as its first element. For each a_i in $\mathcal{A}$ with $i > 0$ holds $T(a, a_i) \geq T(a, a_{i+1})$.

For a final mapping m of artists a to a category in L, we inspect the most occurring category mapped by m' to a and its n nearest neighbors.

$$m(a) = \text{argmax}_{h \in L}\Big(\sum_{0 \leq i \leq n} I(a_i, h)\Big)$$

with

$$I(a_i, h) = \begin{cases} 1 & \text{if } m'(a_i) = h \\ 0 & \text{otherwise.} \end{cases}$$

If two categories have an equal score, we select the first occurring one. That is, the category that is mapped by m' to a or to the artist most related to a.

5. Experiments

We present two experiments of our method in the music domain.

In the first, we categorize a list of 224 artist into genres [8]. Contrary to prior work [8, 12], we do not cluster related artists, but are interested in a mapping of artists to genres. We thus explicitly label each artist with a genre.

In the second experiment we map artists to the mood associated with their music using web data. To our best knowledge, no previous work addresses this issue. Based on data of MoodLogic [10], we assign moods to a list of artists using the method presented.

5.1. Genre categorization

We added all names of artists in the list composed by Knees et al. [8] to the set A. This list consists of 14 genres, each with 16 artists.

The genres mentioned in the list are not all suitable for finding co-occurrences. For example, the term *classical* is ambiguous and *Alternative Rock/Indie* is not a term frequently used. We therefore manually rewrote the names of the genres into unambiguous ones (such as *classical music*) and added some synonyms. After collecting the numbers of co-occurrences between artists and genres, we summed up the scores of the co-occurrences for synonyms. Thus, for each artist a the number of co-occurrences with the terms *Indie* and *Alternative rock* are added to the co-occurrences of a with the genre *Alternative Rock/Indie*.

We performed the experiment three times, using each of the methods described to obtain the co-occurrences.

Motivated by the results in [12], for PCM we used the `allintitle` option. We also added the extra term *music* for finding co-occurrences between the artists. For example the terms *Bush* and *Inner Circle* co-occurred a lot on the

```
"#1 (artists OR bands OR
   acts OR musicians) like #0"
"#1 (artists OR bands OR
   acts OR musicians) such as #0"
"#1 (artists OR bands OR acts
    OR musicians)  for example #0"
"#0 and other #1 (artists
    OR bands OR acts OR musicians)"
```

Table 1. The four patterns for artist (#0) - genre (#1) relation.

```
"like #0 and #1"
"such as #0 and #1"
"including #0 and #1"
"for example #0 and #1"
"namely #0 and #1"
"#0 and #1"
"#0 #1 and other"
```

Table 2. patterns for artist - artist relation.

web, due to American politics. By adding the term *music* we restrict ourselves to documents handling music.

For PM we selected for the genre-artist relations the patterns in Table 1 from a list of patterns found expressing this relation [7]. We considered two artists related, if the two co-occur in some enumeration. Since we are not interested in the nature of the relatedness between artists, we selected general enumeration patterns (Table 2) to obtain co-occurrences.

In Table 3 the performance of the preliminary mapping can be found for the three methods ($n = 0$). We were able to map all artists to a genre. Co-occurrences between genres and artists thus could be found using PCM, PM as well as DM. The latter performs best. With respect to the preliminary mapping, the method with the smallest amount of Google queries performs best. The data found on the best ranked documents is thus reliable.

Using DM only few related artists can be found on the documents visited. This leads to a stable performance for the final mapping when expanding the list of related artists (Figure 1). That is, we only consider artists that co-occur at least once. Contrary to especially PCM, large numbers of n do not deteriorate the precision.

The performance of the pattern-based method strongly improves by considering related artists, the best performance is obtained for $n = 8$. All methods perform best for values of n between 5 and 13. The *Rock n' Roll* artists proved to be the most problematic to categorize. The artists in the genres *classical*, *blues* and *jazz* were all correctly categorized with the best scoring settings.

method	$n = 0$	best	(corresponding n)
PCM	71.4	81.3	(13)
PM	72.2	88.8	(8)
DM	83.9	87.1	(5)

Table 3. Precision (%) without related artists and best precision per method.

With the supervised method discussed in [8] a precision of 87% was obtained using complex machine learning techniques and a relatively large training set. In [12] a precision of up to 85% precision was obtained using $\mathcal{O}(|A|^2)$ Google queries. We can conclude that our simple and unsupervised method produces similar results. A important observation is, that the best scoring methods (document- and pattern-based) are the ones performing only $\mathcal{O}(|A| + |G|)$ Google queries. Our approach is thus well suited for categorizing larger sets of artists.

5.2. Mood categorization

Apart from a genre, music meta-data providers often associate a mood with an artist, song or album. MoogLogic [10] distinguishes seven moods: *upbeat*, *happy*, *sad*, *brooding*, *aggressive*, *sentimental* and *romantic*. Contrary to e.g. AMG [1], with each song only one mood is associated. In this second experiment, we investigate whether the same methods can be used without adaptations to identify the mood of the music of some artist. We here use the assumption that from the list of seven moods a most appropriate one can be assigned to each artist.

Since no prior work is known to us in this field, we have composed two test sets ourselves [1] , based on data from MoodLogic. We conducted the experiment on a set of 230 artists. The setup was equal to the first experiment. We only used different patterns for the identification of co-occurrences between moods and artists (Table 4).

We evaluated the performance using two set sets. The first set contains for each of the seven moods the two artists that corresponded best with a mood. For the second test set,

[1] See `http://gijsg.dse.nl/ismir06`

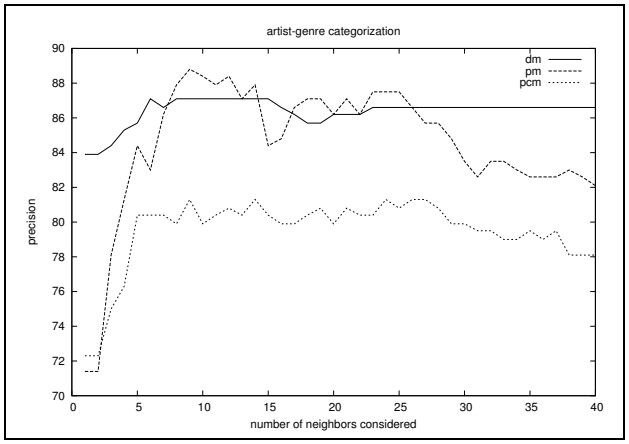

Figure 1. Precision for genre categorization.

"#1 (mood OR tunes OR style) by #0"
"#1 (mood OR tunes OR style) like #0"
"#1 (mood OR tunes OR style) of #0"
"#0's #1"

Table 4. Patterns for the mood (#1) - artist (#0) relation.

we used a less strict criterion. If at least 70% of the songs of an artist was associated with one mood, we added it to the test set. This second test set contains 47 artists.

Figures 2 and 3 show the performance of the three methods in mood categorization. These tests indicate that DM again is the most constant and reliable. It seems that PCM is unsuited for categorizing artists into moods. Although the results are not as convincing as the genre categorization, especially DM performs significantly better than the baseline of 14%, or 1 out of 7 correct.

Even though the data collected by DM is sparse (e.g. no mood could be assigned to R.E.M.), this test shows that this method is the most reliable in mood categorization. The performance of PM can be explained by the observation that an artist and moods from this list occur rarely within a sentence. The hypothesis that taking related artists into account will reduce errors in the categorization only holds for DM. This technique compensates for the artists for which no preliminary mapping could be found.

The overall performance could be improved by adding synonyms for the moods in the list. On the one hand the terms *brooding* and *upbeat* are rare and on the other hand *happy* and *sad* do not always reflect the mood of music. Apparently, the relative scoring function does not compensate for this.

Unlike the model with seven distinct moods used in MoodLogic, multiple moods seem to be applicable to a single artist or even to a single song. Especially the distinction between *upbeat* and *happy* sometimes seems arbitrary. However, we expect a song not to be both *aggressive* and *romantic*. AMG assigns multiple moods to artists, for example 21 of these moods apply to *Madonna*. The document-based method gives rise to research the assignment of multiple moods to songs or artists.

6. Related Work

We observe two areas of related work: research related to web-based relation identification and work on artist classification with web data.

Early work on relation identification from the web can be found in [3]. Brin describes a website-dependent approach to identify hypertext patterns that express some relation. For each web site, such patterns are learned and explored to identify instances that are similarly related. The idea of using patterns for relation extraction is similar to PM, although the extracted relation are not evaluated.

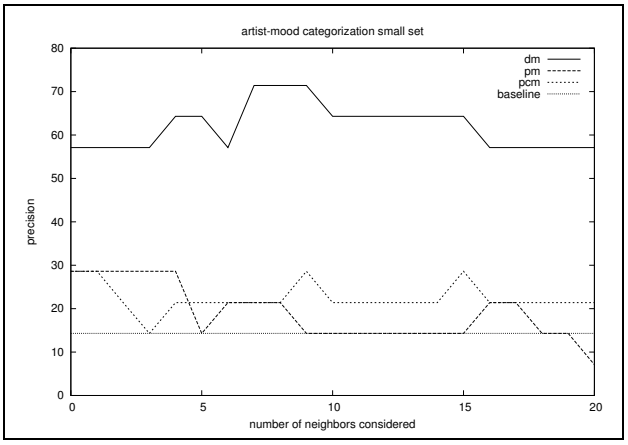

Figure 2. Precision for mood categorization with small set.

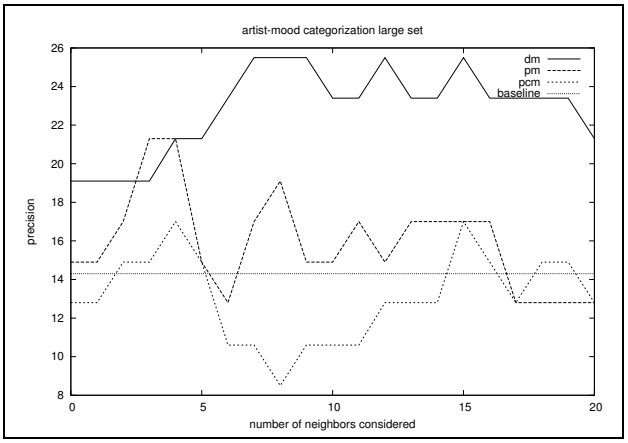

Figure 3. Precision for mood categorization with larger set.

Cimiano and Staab descibe a method to use a search engine to verify a hypothesis relation [6]. For example, if we are interested in the 'is a' or hyponym relation and we have a candidate instance pair (*Nile*, *river*) for this relation, we can use a search engine to query phrases expressing this relation (e.g. *"rivers such as the Nile"*). The number of hits to such queries is used to determine the validity of the hypothesis. Contrary to our method, a majority voting is used. In [11] various techniques are discussed to identify relations between concepts from the web for a question answering system.

In [2] a number of documents on art styles are collected. Names of painters are identified within these documents. The documents are evaluated by counting the number of painters in a training set (of e.g. *expressionists*) that appear in the document. Painters appearing on the best ranked documents then are mapped to the style. This methods differs from DM in two aspects. First we do not collect a constant amount of web pages, but we collect web pages for all elements in the sets A and L. Secondly, De Boer et al. use a training set and page evaluation, where we just observe co-occurrences.

The number of Google *hits* for pairs of terms can be used

to compute a semantic distance between terms [5]. The nature of the relation is not identified, but the technique can for example be used to cluster painters. In [15] a similar method is used to cluster artists using search engine counts.

In [12], the number of Google hits of combinations of artists is used in classifying artists. In PCM we use the same techniques to obtain these figures, but do not use machine learning techniques to interpret them. Moreover, we map artists to categories instead of clustering them. Co-occurrences between artists using search engine hits can also be used to discover prototypical artists per genre [13].

A document based technique in artist classification is described in [8]. For all artists, a number of documents is collected using a search engine. For sets of related artists a number of discriminative terms is learned. These terms are used to classify the other artists using support vector machines. The documents are obtained in a similar way in our document-based method. However, we restrict ourselves to identifying names of artists and categories on the documents.

7. Conclusions and Future Work

We have discussed three alternative methods to obtain co-occurrences between terms using a search engine. These methods are applied to gain a preliminary mapping between artists and categories such as genre. When related artists share the same category, the mutual distance between artists can be used to obtain a more reliable mapping.

The three alternatives used have a different complexity with respect to the number of queries to a search engine. The method using patterns and the one using complete documents are linear to the number of items in the sets of artists, where the page-count-based mapping is quadratic. This distinction is important for categorizing large sets of artists, since search engines allow only a limited amount of automated queries per day.

In the first experiment we showed that we can precisely categorize artists to genres, where the most efficient methods with respect to the Google complexity perform best.

The second experiment consisted of the mapping of artists to the moods associated with their music. This novel approach in artist categorization had encouraging results, but is open to improvement. The document-based method seems best suited to categorize artists into moods.

The two experiments showed that the methods with least amount of queries give the best results. We showed that simple and unsupervised methods can be used for a reliable categorization. Therefore, these methods are well suited to be applied to larger data-sets.

In future work, we want to further explore the field of automatically categorizing music into moods. We will investigate methods to categorize songs or albums rather than artists. Moreover, the moods identified in the experiment are less suited, since the terms are ambiguous. Automatic identification of terms associated with moods (analogous to work in [8]) is an interesting option to explore. Finally, we plan to test the methods using a richer categorization, since we currently only identified a few broad genres and moods. The meta-data on moods provided by AMG can be used as a benchmark.

References

[1] All Music Guide (AMG). website. http://www.allmusic.com.

[2] V. d. Boer, M. v. Someren, and B. J. Wielinga. Extracting instances of relations from web documents using redundancy. In *Proceedings of the Third European Semantic Web Conference (ESWC'06)*, Budva, Montenegro, June 2006.

[3] S. Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at sixth International Conference on Extending Database Technology (EDBT'98)*, 1998.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] R. Cilibrasi and P. Vitanyi. Automatic meaning discovery using Google. http://www.cwi.nl/˜paulv/papers/amdug.pdf.

[6] P. Cimiano and S. Staab. Learning by Googling. *SIGKDD Explorations Newsletter*, 6(2):24–33, 2004.

[7] G. Geleijnse and J. Korst. Learning effective surface text patterns for information extraction. In *Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 1–8, April 2006.

[8] P. Knees, E. Pampalk, and G. Widmer. Artist classification with web-based data. In *Proceedings of 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 517–524, Barcelona, Spain, October 2004.

[9] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[10] MoodLogic. website. http://www.moodlogic.com.

[11] D. Ravichandran. *Terascale Knowledge Acquisition*. PhD thesis, University of Southern California, 2005.

[12] M. Schedl, P. Knees, and G. Widmer. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing (CBMI'05)*, Riga, Latvia, June 2005.

[13] M. Schedl, P. Knees, and G. Widmer. Discovering and Visualizing Prototypical Artists by Web-based Co-Occurrence Analysis. In *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, September 2005.

[14] J. Véronis. Weblog, 2006. http://aixtal.blogspot.com.

[15] M. Zadel and I. Fujinaga. Web services for music information retrieval. In *Proceedings of 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 2004.