MEANINGFULLY BROWSING MUSIC SERVICES

Tim Pohle[1], **Peter Knees**[1], **Markus Schedl**[1] **and Gerhard Widmer**[1,2]
[1)] Dept. of Computational Perception
Johannes Kepler University, Linz, Austria
[2)] Austrian Research Institute for Artificial Intelligence (OFAI)
Vienna, Austria

ABSTRACT

We present a browser application that offers the user an enhanced access the content of music web services. Most importantly, the technique we apply aims at making it feasible to add to the automated suggestion of similar artists some intentional spin, or direction. At the heart of the algorithm, automatically derived artist descriptions are analyzed for common topics or aspects, and each artist is described by the amount it is associated with each of these topics. The browser application enables the user to formulate a query by means of these underlying topics by simply adjusting slider positions. The best matching artist is shown, and its web page found on the web music service is displayed.

1 INTRODUCTION

Today, it is common for online music shops and music web services to have functions for convenient browsing and discovery of new music. One of these functions is the suggestion of similar artists. Such a similar artists function is quite useful, but it has the drawback that the process of discovering new artists may turn into a trial-and-error process, as it may not be immediately obvious if the suggested artists are closer to the user's particular likes (or dislikes). Recently, we have suggested a procedure to describe artists by the aspects (or topics) they are commonly associated with [1]. Based on such a description, it is possible to give the similar artist suggestions an intentional spin, or direction. In the demonstration presented here, we have implemented a browser application that enables the user to access the content of a music web service with this technique. The recommendation tools offered by the music service's web interface are also available, so that the application allows the user to get an overall impression how the new technique could complement existing approaches.

2 RELATED WORK

There are music recommendation services (such as *musiclens.de*) that allow for browsing music by adjusting slider values. The main difference is that the approach presented here automatically derives the browsable categories from the given data, thus no time-consuming manual annotation is necessary. Also, our approach does not work on the track level but on the artist level.

3 ALGORITHM

In this section, we give an overview of the implemented procedure. Details can be found in [1]. The outcome of the procedure when applied to the data used in this demonstration is given in the next section. The outline of the implemented procedure is as follows:

1) Obtain artist descriptions. For each artist that is in the repository, a description is automatically extracted from the web. This is done by querying a web search engine with +"`artist name`" +*music* + *review*. The found pages are analyzed for the occurrence of particular music-related words, and a $TF \times IDF$ vector is calculated [2]. The output of this step is a long list of words associated with each artist. Each of the words has a weight associated.

2) Analyse artist descriptions for common properties. The artist descriptions (long vectors of weighted terms) are compressed to few – ideally meaningful – concepts that make a high-level interaction feasible. For this step, we apply Non-Negative Matrix Factorization (NMF) to yield eight main concepts [3]. The input of NMF are the $TF \times IDF$ vectors of all artists, and the most important output are eight vectors that allow a projection of each high-dimensional artist vector down to eight dimensions.

3) Represent each artist as a mixture of the common properties. This means to apply the transformation calculated in the previous step to obtain a compressed representation for each artist. After this transformation, each artist is described by the amount it is associated with each of the eight concepts.

The last step above yields a vector for each artist. The user query also is represented by a vector of the same length, so it is possible to calculate a similarity between the query vector and each artist by applying the cosine similarity measure. The user query can be seen as a user-generated artist description.

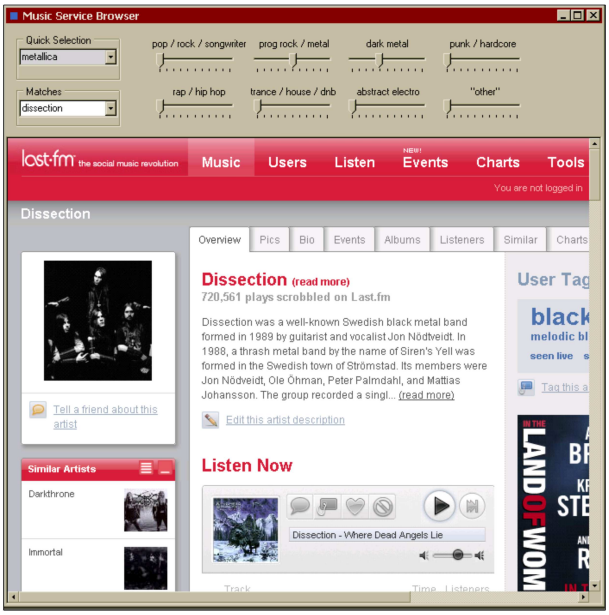

Figure 1. Demonstration: The music service browser. Each artist is represented as a vector giving the amount this artist is associated with each of the categories represented as sliders on the top. When the user selects an artist (via *Quick Selection*), its representation is transferred to the sliders and its *last.fm* web page is shown. When the user modifies the slider positions, the suggested artists (in the *Matches* box) are changed and a different artist's page is shown.

4 IMPLEMENTATION

For the demonstration, we use list of more than 2.000 artist names that have associated web pages on the music web service *last.fm* [1]. For each of these artists, up to 100 web pages are retrieved and analyzed as described in Section 3. The artist data is analyzed for eight concepts. For each of the found concepts, the terms with the highest weights are given in Table 1. In the browser application, each of these categories is represented by a slider that is labelled with a manually defined description of the respective concept (Figure 1).

4.1 Browser Usage

When using the browser, the user may immediately select an artist that is known to her by selecting this artist in the *Quick Selection* box. When doing so, the artist's page on the web service is displayed, and the internal representation of the selected artist is transferred to the slider positions. Then, the user may increase or decrease the influence of each aspect on the artist suggestion. For example, in Figure 1, the chosen seed artist was *Metallica*, and the amount of *Dark Metal* has been increased. The artists that best match the current slider positions are listed in the *Matches* box. When the suggested artists change during a slider adjustment, both the content of the *Matches* box is updated, and the page associated with the the currently best-matching artist is shown.

[1] To obtain the artist names, we used *audioscrobbler.net*

progressive metal	idm	black metal
dream theater	anticon	deicide
power metal	warp	morbid angel
prog metal	rephlex	blackmetal
progressive rock	bjork	dismember

pop rock	composers	rappers
madonna	research	gangsta
rock and roll	theatrical	dr dre
singer songwriter	new york	rap
songwriter	horror	hip hop

trance	screamo
progressive trance	emo
progressive house	punk rock
paul oakenfold	epitaph
breakbeat	hardcore

Table 1. The eight categories that are found for the artists available in the demo application. For each category the five terms with the largest weight are given. In the browser, each category is represented by one slider.

5 CONCLUSIONS

We present a demonstration application that implements a new approach for browsing music services and finding similar artists in a directed way. Based on the demonstration, the user can get an impression how the new technique may complement existing recommendation tools by offering an additional way for high-level interaction with a music repository.

6 ACKNOWLEDGEMENTS

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L112-N04, and by the EU 6th FP project S2S2 ("Sound to Sense, Sense to Sound", IST-2004-03773). The Austrian Research Institute for Artificial Intelligence also acknowledges financial support by the Austrian Federal Ministries BMBWK and BMVIT.

7 REFERENCES

[1] T. Pohle, P. Knees, M. Schedl, and G. Widmer, "Building an interactive next-generation artist recommender," in *Proc.* 5^{th} *CBMI*, 2007.

[2] B. Whitman and S. Lawrence, "Inferring Descriptions and Similarity for Music from Community Metadata," in *Proc. Intern. Computer Music Conference*, 2002.

[3] Wei Xu, Xin Liu, and Yihong Gong, "Document Clustering Based On Non-negative Matrix Factorization," in *Proc. SIGIR03*, Toronto, Canada, 2003.