ON THE USE OF MICROBLOGGING POSTS FOR SIMILARITY ESTIMATION AND ARTIST LABELING

Markus Schedl
Department of Computational Perception
Johannes Kepler University
Linz, Austria
markus.schedl@jku.at

ABSTRACT

Microblogging services, such as *Twitter*, have risen enormously in popularity during the past years. Despite their popularity, such services have never been analyzed for MIR purposes, to the best of our knowledge. We hence present first investigations of the usability of music artist-related microblogging posts to perform *artist labeling* and *similarity estimation* tasks. To this end, we look into different text-based *indexing* models and *term weighting measures*. Two artist collections are used for evaluation, and the different methods are evaluated against data from *last.fm*. We show that microblogging posts are a valuable source for musical meta-data.

1. INTRODUCTION

With the emergence of blogging services, social networks, platforms to share user-generated content and corresponding tags, services for music recommendation and personalized Web radio, such as *last.fm* [12], and in general all services and platforms commonly summarized by the term “Web 2.0”, a new era of Web-based user interaction has started. The term “Web 2.0” was coined in 1999 by DiNucci [5], but did not become popular until 2004, when O’Reilly launched the first Web 2.0 conference [19].

Microblogging is one of the more recent phenomena in the context of the “Web 2.0”. Microblogging services offer their users a means of communicating to the world in real time what is currently important for them. Such services had their origin in 2005, but gained greater popularity not before the years 2007 and 2008 [28]. Today’s most popular microblogging service is *Twitter* [30], where millions of users post what they are currently doing or what is currently important for them. [9]

Despite the enormous rise in usage of microblogging services, to the best of our knowledge, they have not been used for music information extraction and retrieval yet. Hence, in this paper we present first steps towards assessing microblogging posts for the MIR tasks of *music artist labeling* and *similarity measurement*. We will show that even though such data is noisy and rather sparse, results comparable to other text-based approaches can be achieved.

The remainder of the paper presents and discusses related literature (Section 2), elaborates on the methods employed for similarity measurement and artist labeling (Section 3), gives details on the conducted evaluation experiments and discusses their results (Section 4), and finally summarizes the work (Section 5).

2. RELATED WORK

As this work is strongly related to text-based music information extraction and to Web content mining, we are going to review related work on these topics in the context of MIR. The past five years have seen the emergence of various text-based strategies to address MIR tasks, such as automated labeling, categorizing artists according to a given taxonomy, or determining similarities between tracks or artists.

Early work on text-based MIR focused on extracting information from artist-related *Web pages*. Cohen and Fan [4] query search engines to gather music-related Web pages, parse their DOM trees, extract the plain text content, and distill lists of artist names. Similarities based on co-occurrences of artist names are then used for artist recommendation. Web pages as data source for MIR tasks are also used in [7, 32], where the authors rely on a search engine’s results to artist-specific queries to determine artist-related Web pages. From these pages, weighted term profiles, based on specific term sets (e.g., adjectives, unigrams, noun phrases), are created and used for classification and recommendation. Baumann and Hummel [3] extend this work by introducing certain filters to prune the set of retrieved Web pages, aiming at suppressing noisy pages. Another extension is presented in [10] for similarity measurement and genre classification. Knees et al. do not use specific term sets, but create a term list directly from the retrieved Web pages and use the χ^2-test for term selection, i.e., to filter out terms that are less important to describe certain genres. Other Web-based MIR approaches use page count estimates returned by search engines. For example, in [8, 26] co-occurrences of artist names and terms specific to the music domain, as returned by search engine’s page count estimates, are used to categorize artists.

Another category of Web-based approaches to derive artist similarity exploits *user-generated playlists*. For example, in [2] Baccigalupo et al. analyze co-occurrences of artists in playlists shared by members of a Web commu-

Term Set	Cardinality	Description
`all_terms`	681,334	All terms that occur in the corpus of the retrieved *Twitter* posts.
`artist_names`	224	Names of the artists for which data was retrieved.
`dictionary`	1,398	Manually created dictionary of musically relevant terms.
`last.fm_toptags_overall`	250	Overall top-ranked tags returned by *last.fm*'s `Tags.getTopTags` function.
`last.fm_toptags_collection`	5,932	Aggregated top-ranked tags retrieved from *last.fm* for all artists in the collection.
`last.fm_toptags_topartists`	12,499	Aggregated top-ranked tags retrieved from *last.fm* for *last.fm*'s 2,000 top-played artists.

Table 1. List of the term sets used to index the *Twitter* posts. The cardinalities of term sets `all_terms`, `artist_names`, and `last.fm_toptags_collection` are based on the collection *C224a*.

nity. More than one million playlists made publicly available via *MusicStrands* [18] (no longer in operation) were gathered. The authors not only consider the co-occurrence of two artists in a playlist as an indication for their similarity, but also take into account that two artists that consecutively occur in a playlist are probably more similar than two artists that occur farther away from each other.

A recent approach derives similarity information from the *Gnutella* [22] *P2P file sharing network*. Shavitt and Weinsberg [27] collected metadata of shared music files from more than 1.2 million *Gnutella* users.The authors use this data for artist recommendation and song clustering, giving special emphasis to adjusting for the popularity bias.

Another data source related to the "Web 2.0" is *social tags*. [11] gives a good overview of their use in MIR. In [15] a semantic space is built, based on social tags extracted from *last.fm* and *MusicStrands*. The authors use this data for categorizing tracks into mood categories and present a user interface to browse a music collection according to mood. As an alternative to retrieving social tags from music information systems, tags may also be gathered via games designed to encourage their players to assign meaningful descriptions to a music piece [14, 17, 29]. Due to their design, this method can effectively reduce noise.

3. MINING TWITTER POSTS

To acquire user posts we queried *Twitter*'s Web API [31] in February and March 2010 with the names of the music artists under consideration. We downloaded up to 100 posts per query and extracted the plain text content. Earlier work on text-based music information retrieval [10, 26, 32] suggests to enrich the artist names with additional keywords, such as "music review" or "music genre style", to guide the retrieval process towards sources that contain information on music. However, preliminary classification experiments with various additional music-related keywords revealed that this strategy does not work well for *Twitter* posts. Restricting the search with any keyword in addition to the artist name in fact decreases the number of available user posts so strongly that even for the popular artists in our test collection *C224a* (cf. Subsection 4.1) the resulting feature vectors become very sparse.

After having downloaded the *Twitter* posts for each artist, we built an *inverted word-level index* [34] based on a modified version of the *lucene* [16] indexer. To investigate the influence of the term set used for indexing, we built various indexes using the term sets depicted in Table 1. The table further gives the term sets' cardinality. In cases where this cardinality depends on the size of the corpus, the values are based on collection *C224a* (cf. Subsection 4.1). The list denoted as `dictionary` consists of terms that we manually collected from various sources and somehow relate to music. This list resembles the one used in [21] and [24]. Included terms represent, for example, musical genres and styles, locations, instruments, emotions, and epochs.

Term weighting is performed using variants of the *term frequency* (tf) measure and the *term frequency · inverse document frequency* ($tf \cdot idf$) measure [33]. The term frequency $tf_{t,a}$ is the total number of occurrences of term t in all *Twitter* posts retrieved for artist a. The $tf \cdot idf_{t,a}$ function is defined as follows, where n is the total number of artists and df_t is the number of artists whose retrieved posts contain t at least once:

$$tf \cdot idf_{t,a} = \ln\left(1 + tf_{t,a}\right) \cdot \ln\left(1 + \frac{n}{df_t}\right) \tag{1}$$

The basic idea of the $tf \cdot idf_{t,a}$ measure is to increase the weight of t if t occurs frequently in the posts retrieved for a, and decrease t's weight if t occurs in a large number of posts retrieved for *different* artists and is thus not very discriminative for a.
Since we are not interested in individual *Twitter* posts, but rather in a document describing a certain music artist, we aggregate all posts retrieved for an artist into a virtual document, based on which the term weights are calculated.

3.1 Similarity Estimation

Based on the term weighting vectors, we derive similarity between artists by applying the *cosine similarity measure* [23]. The cosine measure normalizes the data in that it accounts for different document lengths. To this end, only the angle between the weight vectors in the feature space is considered. In our case, the virtual documents for two artists a and b may be of very different length (depending on the number and length of the corresponding posts), which is likely to distort the weighting. [1] Therefore, we apply the cosine similarity measure between the $tf \cdot idf$ vectors of each pair of artists (a, b) according to Formula 2, where $|T|$ is the cardinality of the term set, i.e., the dimensionality of the term weight vectors. θ gives the

[1] The fact that usually much more data is available for popular artists than for lesser known ones, and the resulting likely distortion of results, is commonly referred to as "popularity bias".

angle between a's and b's feature vectors in the Euclidean space.

$$sim(a,b) = \cos\theta = \left(\frac{\sum_{t=1}^{|T|} tf \cdot idf_{t,a} \cdot tf \cdot idf_{t,b}}{\sqrt{\sum_{t=1}^{|T|} tf \cdot idf_{t,a}^2} \cdot \sqrt{\sum_{t=1}^{|T|} tf \cdot idf_{t,b}^2}} \right) \quad (2)$$

3.2 Labeling

A good similarity estimation function is crucial for many application areas of MIR techniques, for example, to build recommender systems, to generate intelligent user interfaces via clustering, or for automated playlist generation. Another related MIR task is automatically assigning labels/descriptors to an artist or a song. This allows to perform categorization of artists or songs into certain classes, for example, mood categories or a genre taxonomy.

We were interested in analyzing the potential of user-generated *Twitter* posts to perform automated categorization or labeling of music artists, also known as "autotagging" [6]. To this end, we compiled a list of *last.fm*'s top tags for the top artists (56,396 unique terms) and subsequently indexed the *Twitter* posts, taking this list as dictionary for our modified *lucene* indexer. Employing either the tf or the $tf \cdot idf$ measure, we used the top-ranked terms of each artist to generate labels.

4. EVALUATION

4.1 Test Collections

To compare the results of the proposed approaches to existing methods, we first ran evaluation experiments on the collection presented in [10]. It comprises 224 well-known artists, uniformly distributed across 14 genres. We will denote this collection as *C224a* in the following.

Since we further aim at evaluating the approaches on a real-world collection, we retrieved the most popular artists as of the end of February 2010 from *last.fm*. To this end, we used *last.fm*'s Web API [13] to gather the most popular artists for each country of the world, which we then aggregated into a single list of 201,135 artist names. Since *last.fm*'s data is prone to misspellings or other mistakes due to their collaborative, user-generated knowledge base, we cleaned the data set by matching each artist name with the database of the expert-based music information system *allmusic.com* [1]. Starting this matching process from the most popular artist found by *last.fm*, and including only artist names that also occur in *allmusic.com*, we eventually are given a list of 3,000 artists. We will denote this collection, which is used for artist labeling, as *3000a*.

4.2 Similarity Estimation

While the authors are well aware of the fact that "genre" is an ill-defined concept and that genre taxonomies tend to be highly inconsistent [20], we unfortunately do not have access to reliable and comprehensive similarity data, against which we could perform comparison. We therefore opted for a genre classification task that serves as a proxy for similarity measurement. We employed a *k-nearest neighbor* classifier (leave-one-out), and we investigated classification accuracy for different values of k, different term sets used for indexing, and different term weighting measures (tf and $tf \cdot idf$). We ran the classification experiments on collection *C224a*, since this artist set is already well-established in the literature, and results are therefore easy to compare.

4.2.1 *Results*

Figure 1 shows a detailed illustration of the k-NN classification results for different term sets and term weighting measures, using collection *C224a*. In general, $tf \cdot idf$ works better for the task of similarity estimation than the single tf value. The best classification results achievable using $tf \cdot idf$ are 72.52% accuracy with `all_terms` and a 9-NN classifier and 72.38% accuracy with an 8-NN-classifier and `last.fm_toptags_collection`.

Interestingly, the tf-based predictors (which, in general, perform worse than the $tf \cdot idf$-based predictors), perform comparable to the best $tf \cdot idf$-based classifiers when using `artist_names` for indexing. This setting resembles the co-occurrence approach described in [25], where accuracies of 54% and 75% (depending on the query scheme) were achieved for collection *C224a*. Using tf-weighting, our approach achieves a maximum of 65.34% accuracy with a 5-NN classifier. The authors of [10] report accuracy values of up to 77% using a k-NN classifier and up to 85% using a *Support Vector Machine* (SVM).

As for the different term sets used for indexing, using all terms in the corpus of *Twitter* posts (term list `all_terms`) yields the best classification results, but is computationally most complex. Using `artist_names` for indexing does not significantly reduce classification accuracy, while remarkably decrease space and time complexity. The good performance of the `artist_names` set can be explained by many *Twitter* posts containing lists of currently listened or favored artists. Such data therefore reveals information on personal playlists.

To investigate which genres tend to be confused with which others, Figures 2 and 3 show confusion matrices of the two best performing approaches. Using `all_terms` (Figure 2), "Folk" artists are often confused with "Country" artists, "Alternative Rock/Indie" performers are frequently predicted to make "Metal" music, and "Rock 'n' Roll" is often predicted for artists performing "RnB/Soul". Using `last.fm_toptags_collection` (Figure 3), the most frequent confusions are "Electronic" artists predicted as "Rap" artists and "RnB/Soul" artists mistaken for "Rock 'n' Roll" artists.

While some confusions are easy to explain, for example, "Country" and "Folk" music is pretty close and in some taxonomies even considered one genre, others are likely only the result of users' preference relations instead of similarity relations. For example, the co-occurrence of two artists (one from genre "Electronic", the other from "Rap") in a user's post may not necessarily indicate that these artists are similar, but that they are similarly liked or played together by the user.

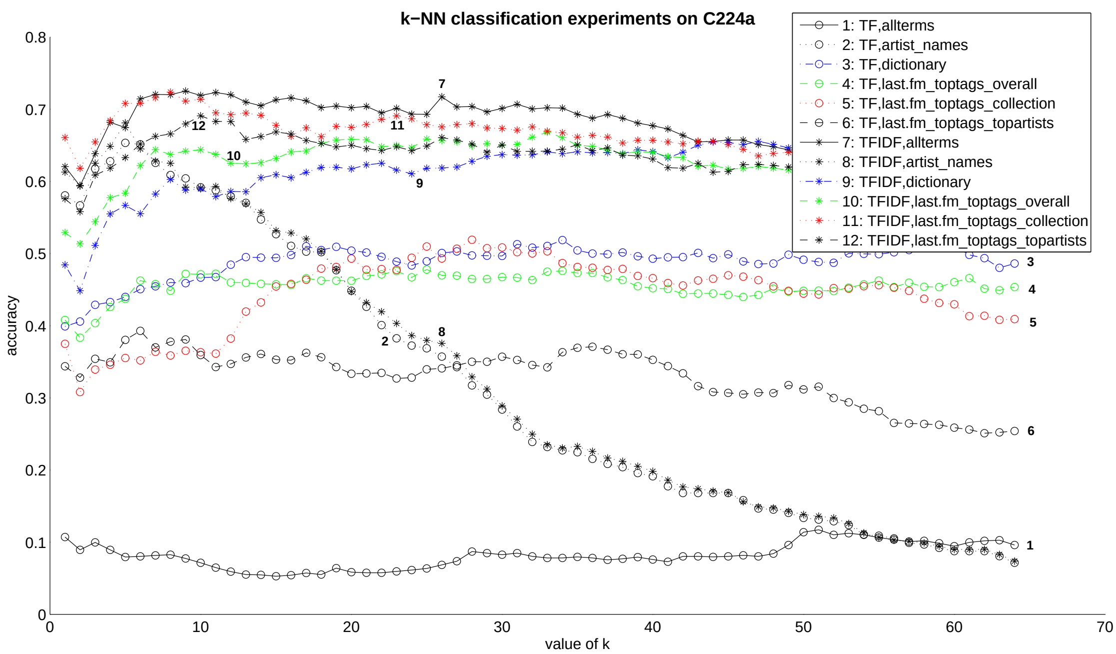
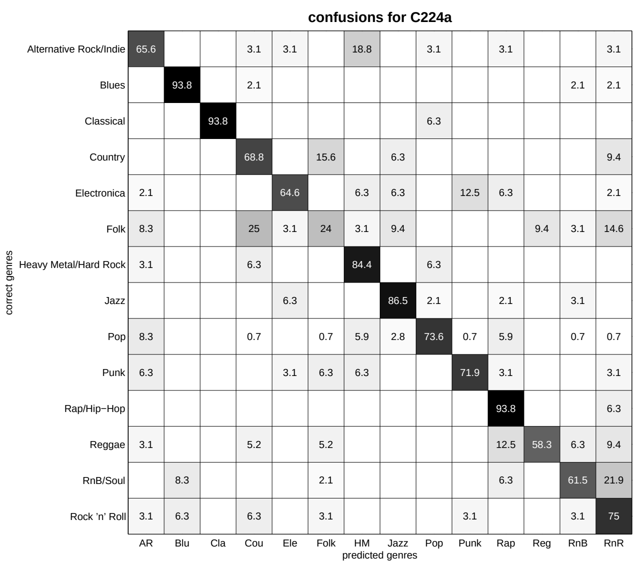


Figure 1. Results of the genre-classification-experiments for different k values of the k-NN classifier, using *C224a*.



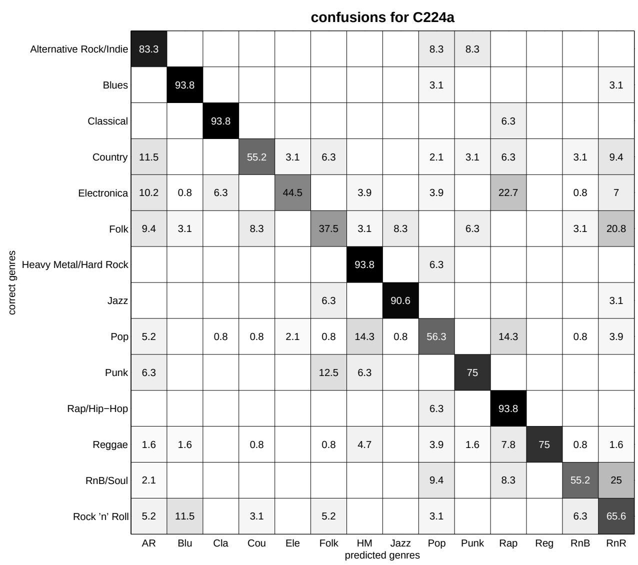


Figure 2. Confusion matrix for the 9-NN classifier on the *C224a* collection using the term list `all_terms`.

Figure 3. Confusion matrix for the 8-NN classifier on the *C224a* collection using the term list `last.fm_toptags_collection`.

4.3 Labeling

To assess the performance of *Twitter* posts for the task of labeling artists, we use an artist a's top-ranked terms (according to each term weighting measure), to predict labels for a. To this end, we index the posts using term list `last.fm_toptags_overall` and a list of tags extracted from *last.fm* for several thousands top-played artists. In total, 56,396 unique terms were obtained.

For evaluation we compare the top-ranked N labels from *Twitter* (according to the term weighting measure) with the top-ranked N tags from *last.fm*. To this end, we calculate an *overlap score* between the two term sets. Aggregating this score over all artists in the collection reveals the average percentage of overlapping terms, considering different quantities N of top-ranked terms. More formally, the *overlap@top-N* is calculated according to Formula 3, where A denotes the artist set, $\#artists_N$ is the number of artists with at least N terms assigned, and $overlap_{tw,fm,a,N}$ is the number of terms in *Twitter*'s set of top-N terms for

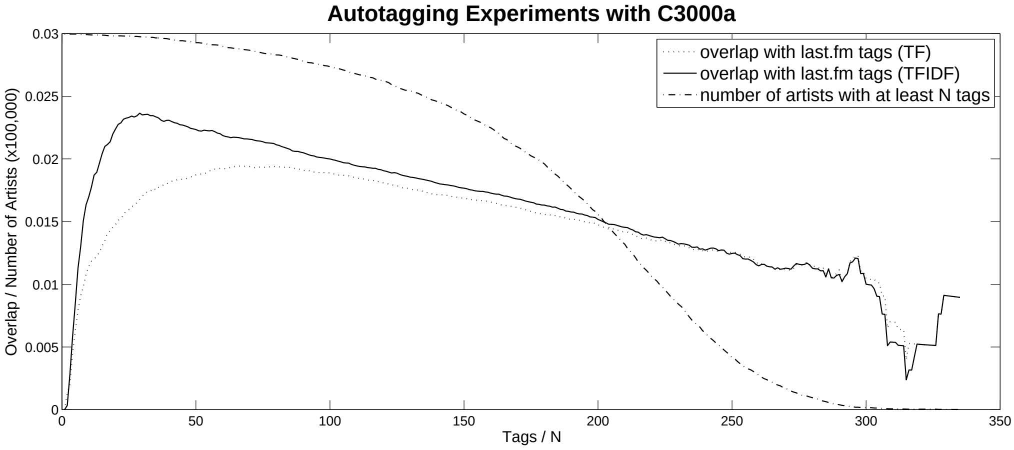


Figure 4. Results of the labeling experiments using *C3000a* and the set of 56,396 tags.

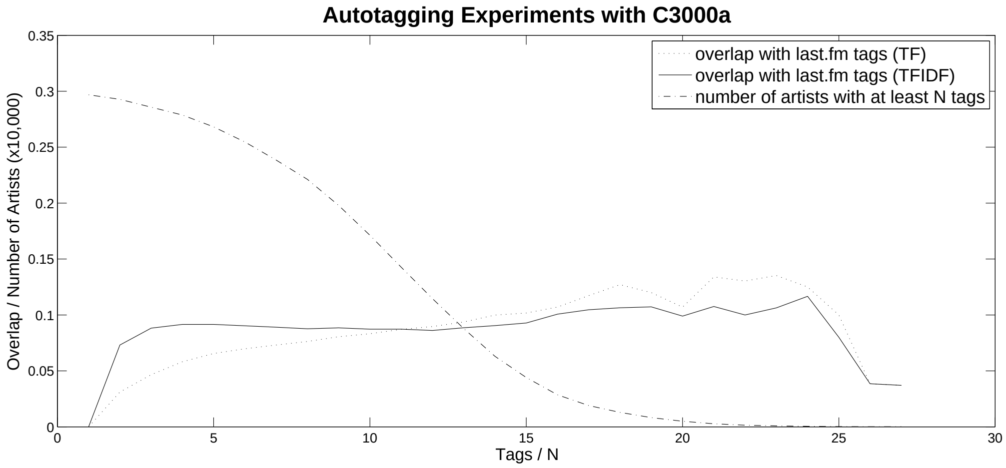


Figure 5. Results of the labeling experiments using *C3000a* and the term set `last.fm_toptags_overall`.

artist a that also occur in *last.fm*'s set of top-ranked tags for a.

$$overlap@top{-}N = \frac{\sum_{a \in A} \frac{overlap_{tw,fm,a,N}}{N}}{\#artists_N} \qquad (3)$$

4.3.1 Results

Figures 4 and 5 show the aggregated overlap scores for collection *C3000a* at different levels of top-N terms/tags using the term set of 56,396 tags and the term set `last.fm_toptags_overall`, respectively. The dash-dotted line reveals the number of artists with at least N terms assigned. The solid line gives the overlap score using $tf \cdot idf$ for term weighting, whereas the dotted line gives the score using tf-weighting.

The low maximum overlap of 2.36% for the 56,396-tag-set ($tf \cdot idf$) is likely caused by a large amount of noise in the *last.fm* tags. Using `last.fm_toptags_overall`, the maximum overlap scores are 13.53% (tf) and 11.67% ($tf \cdot idf$). Taking into account that this is a very challenging task (an overlap of 100% for a certain level of N would mean that the top-N terms according to the *Twitter* posts correspond exactly to the top-N tags from *last.fm* for all artists), these results are better than the sole numbers suggest.

The corresponding maximum overlap scores for collection *C224a* using the 56,396-tag-set amount to 6.68% ($tf \cdot idf$) and 5.39% (tf). Term set `last.fm_toptags_overall` yields maximum overlap scores of 16.36% ($tf \cdot idf$) and 15.22% (tf).

5. CONCLUSIONS AND OUTLOOK

We have shown that *Twitter* posts provide a valuable data source for music information research. In particular for the task of similarity measurement on the artist level, classification results resemble the ones achieved with other text-based approaches using community or cultural data sources, e.g., [10, 25], on the same artist set. For the task of automated labeling, in contrast, only weak to medium overlaps between *Twitter* posts and *last.fm* tags could be determined.

As part of future work, we would like to analyze the localization capabilities of the *Twitter* API. Provided sufficient accuracy, additional geographic data could be used, for example, to spot the most popular artists within a region or country. Successively, such information may be used to reveal the spreading of listening trends around the world. Using geolocation information may also help building country-specific or culture-specific models of music similarity.

6. ACKNOWLEDGMENTS

This research is supported by the *Austrian Fonds zur Förderung der Wissenschaftlichen Forschung* (FWF) under project numbers L511-N15 and Z159.

7. REFERENCES

[1] http://www.allmusic.com (access: January 2010).

[2] Claudio Baccigalupo, Enric Plaza, and Justin Donaldson. Uncovering Affinity of Artists to Multiple Genres from Social Behaviour Data. In *Proc. of ISMIR*, 2008.

[3] Stephan Baumann and Oliver Hummel. Using Cultural Metadata for Artist Recommendation. In *Proc. of WEDELMUSIC*, 2003.

[4] William W. Cohen and Wei Fan. Web-Collaborative Filtering: Recommending Music by Crawling The Web. *WWW9 / Computer Networks*, 33(1–6):685–698, 2000.

[5] Darcy DiNucci. Fragmented Future. *Design & New Media*, 53(4), 1999.

[6] Douglas Eck, Thierry Bertin-Mahieux, and Paul Lamere. Autotagging Music Using Supervised Machine Learning. In *Proc. of ISMIR*, 2007.

[7] Daniel P.W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The Quest For Ground Truth in Musical Artist Similarity. In *Proc. of ISMIR*, 2002.

[8] Gijs Geleijnse and Jan Korst. Web-based Artist Categorization. In *Proc. of ISMIR*, 2006.

[9] Andy Kazeniac. Social Networks: Facebook Takes Over Top Spot, Twitter Climbs. http://blog.compete.com/2009/02/09/facebook-myspace-twitter-social-network (access: March 2010).

[10] Peter Knees, Elias Pampalk, and Gerhard Widmer. Artist Classification with Web-based Data. In *Proc. of ISMIR*, 2004.

[11] Paul Lamere. Social Tagging and Music Information Retrieval. *Journal of New Music Research: From Genres to Tags – Music Information Retrieval in the Age of Social Tagging*, 37(2):101–114, 2008.

[12] http://last.fm (access: March 2010).

[13] http://last.fm/api (access: March 2010).

[14] E. Law, L. von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A Game for Music and Sound Annotation. In *Proc. of ISMIR*, Vienna, Austria, September 2007.

[15] Mark Levy and Mark Sandler. A semantic space for music derived from social tags. In *Proc. of ISMIR*, Vienna, Austria, September 2007.

[16] http://lucene.apache.org (access: February 2010).

[17] Michael I. Mandel and Daniel P.W. Ellis. A Web-based Game for Collecting Music Metadata. In *Proc. of ISMIR*, Vienna, Austria, September 2007.

[18] http://music.strands.com (access: November 2009).

[19] Tim O'Reilly. What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software. http://oreilly.com/web2/archive/what-is-web-20.html (access: March 2010).

[20] François Pachet and Daniel Cazaly. A Taxonomy of Musical Genre. In *Proc. of RIAO*, 2000.

[21] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Hierarchical Organization and Description of Music Collections at the Artist Level. In *Proc. of ECDL*, 2005.

[22] Matei Ripeanu. Peer-to-Peer Architecture Case Study: Gnutella Network. In *Proc. of IEEE Peer-to-Peer Computing*, 2001.

[23] Gerard Salton. The Use of Citations as an Aid to Automatic Content Analysis. Technical Report ISR-2, Section III, Harvard Computation Laboratory, Cambridge, MA, USA, 1962.

[24] Markus Schedl and Peter Knees. Investigating Different Term Weighting Functions for Browsing Artist-Related Web Pages by Means of Term Co-Occurrences. In *Proc. of LSAS*, 2008.

[25] Markus Schedl, Peter Knees, and Gerhard Widmer. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proc. of CBMI*, 2005.

[26] Markus Schedl, Tim Pohle, Peter Knees, and Gerhard Widmer. Assigning and Visualizing Music Genres by Web-based Co-Occurrence Analysis. In *Proc. of ISMIR*, 2006.

[27] Yuval Shavitt and Udi Weinsberg. Songs Clustering Using Peer-to-Peer Co-occurrences. In *Proc. of AdMIRe (IEEE ISM)*, 2009.

[28] http://www.sysomos.com/insidetwitter (access: March 2010).

[29] D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet. A Game-based Approach for Collecting Semantic Annotations of Music. In *Proc. of ISMIR*, Vienna, Austria, September 2007.

[30] http://twitter.com (access: February 2010).

[31] http://apiwiki.twitter.com/Twitter-API-Documentation (access: February 2010).

[32] Brian Whitman and Steve Lawrence. Inferring Descriptions and Similarity for Music from Community Metadata. In *Proc. of ICMC*, 2002.

[33] Justin Zobel and Alistair Moffat. Exploring the Similarity Space. *ACM SIGIR Forum*, 32(1):18–34, 1998.

[34] Justin Zobel and Alistair Moffat. Inverted Files for Text Search Engines. *ACM Computing Surveys*, 38:1–56, 2006.