Exploiting Twitter's Collective Knowledge for Music Recommendations*

Eva Zangerle, Wolfgang Gassler, Günther Specht
Databases and Information Systems, Institute of Computer Science
University of Innsbruck, Austria
{firstname.lastname}@uibk.ac.at

ABSTRACT

Twitter is the largest source of public opinion and also contains a vast amount of information about its users' music favors or listening behaviour. However, this source has not been exploited for the recommendation of music yet. In this paper, we present how Twitter can be facilitated for the creation of a data set upon which music recommendations can be computed. The data set is based on microposts which were automatically generated by music player software or posted by users and may also contain further information about audio tracks.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Performance, Human Factors, Experimentation

Keywords

Recommender Systems, Music Recommendation, Twitter

1. INTRODUCTION

Throughout the last years, music recommendation services have become very popular in both academia and industry. The goal of such services is the recommendation of suitable music for a certain user. This is traditionally accomplished by (i) either taking the user profile consisting of the tracks the user listened to in the past and (if available) the user's rating for songs into account or (ii) analysing the song itself and using the extracted features in order to find similar songs. For the recommendation of music, huge corpora and user profiles are required as there are millions of different audio tracks. There are some large services, such as last.fm[1], which own such big corpora. However, most of them are not publicly available. Especially for academic purposes, only few (mostly small) data sets for the evaluation of the proposed approaches are available, like e.g. the million song data set [4].

Twitter is a publicly available service, which holds huge amounts of data and is still growing tremendously. Twitter stated that there are about 140 million new messages a day. Such messages can also be exploited in the context of music recommendations. Many audio players offer the functionality of automatically posting a tweet containing the title and artist of the track the user currently is listening to. These tweets traditionally contain keywords like `nowplaying` or `listeningto`, like e.g. in the tweet "`#nowplaying Tom Waits-Temptation`". For users who frequently make use of such a service, the set of these tweets can be seen as a user profile in terms of her musical preferences and provide well suited data for e.g. a music recommendation corpus.

In this paper we present an approach for gathering such data and refining it such that the tweeted artists and tracks can directly be related to the free music databases FreeDB and MusicBrainz. As a use case scenario, we present the recommendation of music based on the data set.

This paper is structured as follows. Section 2 describes the processes underlying the creation of the proposed data set. Section 3 features the approach for the recommendation of suitable music tracks as a use case for the gathered data. Section 4 contains related work and Section 5 concludes the paper and discusses future work.

2. DATA SET CREATION

The goal of this approach is the creation of a corpus of music tracks gathered from tweets of users. These tweets contain tracks the user previously listened to and tweeted about (the so-called user stream). In particular, we propose to make use of tweets which have been posted by users or audio players and contain the title and artist of the music track currently played, like e.g. "`#NowPlaying Best Thing I Never Had by Beyonce`". The following sections describe the steps taken for the creation of the data set.

2.1 Crawling of Twitter Data Set and Analysis

The data set was crawled via the Twitter Streaming API-between July 2011 and February 2012. The only publicly available access method is the Spritzer access which only provides real-time access to about 1% of all posted Twitter

*This research was partially funded by the University of Innsbruck (Nachwuchsförderung 2011).

[1] `http://www.last.fm`

WWW Lyon, France, 2012

messages. Due to these restrictions, we crawled 4,734,014 tweets containing one of the keywords `nowplaying`, `listento` or `listeningto` posted by 864,736 different users. This implies an average of 5.5 tweets for each user. Within our data set, the distribution of tweets per user resembles a longtail distribution, as can be seen in Table 1. Such a distribution implies that considering the fact that recommendations can only be made if a user has posted about two or more tracks, a total of 457,675 users and the respective tweets can not be facilitated for our approach as only one tweet of these users is featured within the data set.

Tweets in stream	Users
1	457,675
>3	196,422
>5	126,783
>10	63,017
>100	3,190
$>1{,}000$	253
$>10{,}000$	5

Table 1: Population of User Streams

In total, 5,916,294 hashtags were used within the data set. Clearly due to our used search keywords the hashtags `#nowplaying` and `#listeningto` were the most prominent hashtags within the crawled data set. Also, general hashtags like e.g. `#music`, `#radio` or `#video` have been used frequently. Music streaming services or online radios also make use of hashtags when tweeting about the currently playing track (e.g. `#cityfm` or `#fizy`).

A total of 1,413,983 tweets (29.8% of the whole corpus) featured hyperlinks. An analysis of these URLs revealed that URLs are mostly used to point to music services like e.g. Youtube or Spotify, an online music streaming service. A large part of the hyperlinks lead to the website of the service which was used to post the track information on Twitter, like e.g. tweetmylast.fm or tinysong.com.

2.2 Resolution of Twittered Tracks

This task aims at parsing the gathered tweets and recognizing the artist name and track title mentioned in the tweet. Consider e.g. the tweet "`#NowPlaying Best Thing I Never Had by Beyonce`". For this tweet, we have to extract Beyonce as the artist and "Best Thing I Never Had" as the title of the audio track and match it with a reference music database. Most of the crawled messages are very noisy and consist of many terms which are not concerned with the music track itself. Considering e.g. the tweet "`listening to Hey Hey My My (Out Of The Blue) by Neil Young on @Grooveshark: #nowplaying #musicmonday http://t.co-/7os3eeA`" which contains further information about the online radio service, a URL and other information which are not related to the music track. Especially when dealing with such noisy tweets, the matching is a crucial task as the quality of the data resulting from this step significantly influences the quality of the resulting recommendations.

2.2.1 *Resolution Approach*

As a reference database for artists and the according tracks, we made use of the publicly available databases FreeDB[2] and MusicBrainz[3]. FreeDB contains information about more than 37 million audio tracks, roughly 3,000,000 discs and 766,909 different artists. MusicBrainz was also considered as a reference database as we expected it to be of higher quality than FreeDB. MusicBrainz contains about 8 million tracks of about 650,000 different artists.

The goal of this task is to assign each tweet a FreeDB and a MusicBrainz entry which represents the title and the according artist extracted from the tweet. We tackle this resolution task by making use of a Lucene fulltext index as it allows a simple matching of strings, namely the tweet and a certain FreeDB or MusicBrainz entry. The fulltext index is filled with a combined string containing both the artist and the title of all tracks within the reference databases.

In a next step, we query this fulltext index for each of the tweets within the data set in order to obtain the most suitable FreeDB/MusicBrainz candidates for the title and artist of the track. We then use the top-20 search results of Lucene as candidates for the assignment of tracks to the information mentioned in the according tweet. Lucene's ranking function is based on the term frequency/inverse document frequency measure (tf/idf). This measure is dependent on the length of the query which is not favourable in our approach as tweets contains a high degree of noise (e.g. URLs, feelings, smilies, etc.) which are not part of a track title but also part of the query (the tweet). Therefore, we implemented a bag-of-words similarity measure between the query and the documents contained within the Lucene index similar to the Jaccard similarity measure. Our proposed similarity measure is defined by the ratio between the size of the (term-) intersection of the query and the track and the number of terms contained in the track, as can be seen in Equation 1.

$$sim_{music}(tweet, track) = \frac{|tweet \cap track|}{|track|} \tag{1}$$

The advantage of such a measure is the independence of the length of the query and the reduced influence of the noise in tweets. Furthermore, as our goal is to find the best matching audio track for all given tweets, it is crucial that most terms within the track are matched. However, in the case of multiple search results having obtained an equal score, we still rely on the tf/idf values computed by Lucene. Our proposed score is used for a ranking of the Lucene search results. For each of the tweets, the track which obtained the highest score are assigned to the tweet. In order to be able to set a certain threshold for the scores of the matching entries later, we also store the computed sim_{music}-score.

2.2.2 *Evaluation of Resolution*

For the evaluation of the resolution and the comparison of FreeDB and MusicBrainz, we created a ground truth data set which consists of 100 tweets randomly chosen from the data set. Subsequently, we tried to assign matching tracks in the FreeDB and MusicBrainz databases manually. This task was done by the same person for both reference databases and also contains the resolution of abbreviations or mentions which link to the artist's Twitter account. For example the tweet `#nowplaying @Lloyd_YG ft. @LilTunechi - You` can be resolved to the two Twitter accounts *Lloyd-Young Goldie* and *Lil Wayne WEEZY F* and therefore to the MusicBrainz entry *Lil Wayne feat. Lloyd - You.* Having gathered all possible information from the tweet, the assigning person searched for matching tracks in the database.

[2]`http://www.FreeDB.org`
[3]`http://www.MusicBrainz.org`

If the artist or the title of the track were not directly recognizable in the tweet, single words are used to search the database and find matching artists or titles. We only considered tweets which were resolved to both the according track and artist. Tweets such as `Chris Duarte, famous blues musician - free videos here: http://t.co/UZMXaGQ #blues #guitar #music #roots #free #nowplaying #musicmonday` which only contain information about the artist were not counted as a match. However, such information is also very valuable as it describes the musical taste of a user. For our ground truth data set, we were able to manually assign 57 tracks of FreeDB and 59 tracks of MusicBrainz. This shows that the size of both data sets is similar, however the FreeDB data set is very noisy (typos, spelling errors and variations).

Subsequently we ran our automated Lucene based resolution process on the ground truth dataset using both reference databases (see details in Table 2). Considering a sim_{music}-score threshold of 0.8 we were able to resolve 73% of the ground truth correctly and had an error rate (false positives) of about 10% of all matched tracks. The high number of false positives using the FreeDB data set can be lead back to the noisy entries in FreeDB.

RefDB	Manually	Automated	False Pos.
MusicBrainz	59	43 (73%)	5 (10%)
FreeDB	57	31 (54%)	18 (36%)

Table 2: Resolution Ground Truth (100 tweets)

Due to these obtained results we used MusicBrainz for all further computations (e.g. music recommendations).

3. MUSIC RECOMMENDATIONS

As a use case, we implemented a music recommendation service on top of the data set. The necessary steps for a recommendation of music are described in the following.

The proposed approach for the recommendation of music titles relies on the co-occurrence of titles within a user stream. Based on the obtained tweets and the assigned tracks, we propose to use association rules [2] in order to be able to model the co-occurrence of items efficiently. In the case of the co-occurrence of tweeted music titles, an association rule $t_1 \rightarrow t_2$ describes that a particular user who tweeted about song t_1 also tweeted about song t_2. These rules are the basis for the further recommendation process and are stored as triples $r = (t_1, t_2, c)$, where t_1 and t_2 are tracks which have been tweeted by the same user. c is a variable holding the popularity of the rule. Hence, such a rule denotes that track t_1 and track t_2 both have been listened by c users.

3.1 Ranking of Recommendation Candidates

In this step, the computed association rules are analysed and so-called recommendation candidates are extracted. Based on the rules, the recommended tracks for a certain user are computed by selecting a subset $\mathcal{C} \subseteq \mathcal{T}$ of track recommendation candidates by determining all rules which feature tracks occurring on the user stream. The final step for the recommendation of tracks is the ranking of the recommendation candidates within the set $\mathcal{C}$. Therefore, we make use of the count value c describing the popularity of a certain track within all association rules matching the tracks of the input user stream. Hence, all recommendation candidates are ranked by the respective count values where a higher count value results in a higher rank for the candidate.

3.2 Offline Evaluation

As a first evaluation we performed an offline evaluation and compared the computed track recommendations with recommendations provided by the last.fm API[4] which lists tracks similar to a given track including a score stating the relevance of the song (matching score).

We made use of the MusicBrainz data set as it contains cleaner data than FreeDB. Firstly, we removed all tweets of users who contributed only one tweet and which were matched with a MusicBrainz track with $sim_{music} < 0.8$ to dismiss uncertain mappings. Hence our final data set consisted of 2.5 million tweets of 525,751 users. Based on this data set we computed the according association rules and obtained 500 million distinct rules. Due to computability reasons and API limitations, we chose a subset consisting of the most popular tracks and according rules which are present more than 10 times ($c > 10$). The final data set consisted of 15,000 unique tracks and 90 million distinct rules.

We called the last.fm API for all tracks and the API was able to recognize 13,138 out of 15,000 songs. The API returned 3.2 million similar tracks which we matched with our internal MusicBrainz database. In total, 83% of all tracks with a score > 0.8 were matched. We transformed the gathered last.fm data to association rules and computed the overlap of rules with our rule set. 19% of the last.fm rules are covered by the Twitter-based rules. If we consider only similar tracks of last.fm with a matching score (gathered via the last.fm API) higher than 0.6, the twitter-based rules cover 79% of all rules in the set. When comparing the top-10 recommendations on both sides the coverage is only about 1% of all rules. These low numbers can be lead back to the restrictions of the Twitter API and the resulting sparse data set. Especially the incomplete user profiles decrease the coverage. E.g. within the "taste" subset of the million song data set roughly 70% of the tracks were played more than 10 times. In contrast, in our data set only 5% of the tweets were contained more than 10 times. This fact strengthens the evidence that the crawled data set is not representative enough which can be lead back to the API limitation and uncertainties in the matching processes. Furthermore, due to the diversity of music tracks, such an offline evaluation may not reveal the full potential of the approach. Online evaluations may achieve better results for our proposed approach and are subject to future work.

4. RELATED WORK

Research related to the presented approach can be categorized into (i) approaches dealing with recommendations either for Twitter or based on tweets and (ii) approaches mainly dealing with the recommendation of music.

The utilization of a corpus of tweets for the recommendation of resources has been a popular research topic. For example the recommendation of suitable hashtags is discussed in [14]. Many approaches aim at the recommendation of users who might be interesting to follow, like e.g. in [7]. Such approaches are typically based on the social ties of a user (his followees and followers). There are also many ap-

[4] `http://www.last.fm/api`

proaches which exploit these ties to recommend resources, such as websites [6] or news [12].

As for the second category of related work, the recommendation of music, many different approaches have been presented. Celma [5] provides an overview about this topic. Within Recommender Systems, in principle two major approaches are distinguished [1]: content-based recommendations and collaborative filtering (CF) approaches. Content-based recommendation systems aim at recommending resources which are similar to the resources the user already consumed or showed interest in. Collaborative filtering approaches aim at finding users with a profile similar to the current user in order to recommend items which these similar users also were in favor of. This categorization also holds within music recommendations. Content-based methods for music titles typically rely on the extraction and analysis of audio features. The presented approach relies on the second type as the computation of association rules based on user profiles can be assigned to the class of CF approaches.

However, for music recommendations also a third important aspect is exploited for the computation of recommendations: context. The notion of context has e.g. been defined by Schmidt et al. as being threefold: physical environment, human factors and time [13]. These three factors have all been addressed by music recommendation research. As for the physical environment of a user, e.g. Kaminskas and Ricci presented a location-aware approach for music recommendations [8]. The mood of users has been incorporated for the computation of recommendations in [9] and Baltrunas et al. [3] considered temporal facts when recommending music.

Many approaches exploited user profiles in social networks to recommend resources. Mesnage et al. [10] showed that people prefer the music that their friends in the social network prefer. The Serendip.me project[5] provides its users with music which is selected solely based on the Twitter ties (the followees) of the user. The dbrec project [11] is concerned with recommending music based on the DBPedia data set. In particular, the authors developed a distance metric for resources within DBPedia which enables the authors to recommend similar artists.

However, to the best of our knowledge there are no approaches concerned with the recommendation of music based on an analysis of "nowplaying" user streams on Twitter.

5. CONCLUSION AND FUTURE WORK

In this paper we showed that tweets can be exploited to build a corpus for music recommendations. The comparison with the recommendation service of last.fm showed that despite the sparse corpus due to Twitter's API limitations, the coverage of last.fm's recommendations is up to 79%. The results are very promising although the approach has to be enhanced to be usable in real-world recommendation environments. A mayor improvement would be the expansion of the data set as currently the corpus is very sparse and the user profiles are incomplete. Also, the matching task of noisy tweets deteriorates the quality of recommendations. This is due to the fact that many uncertain matching results have to be dismissed and hence, the size of the usable data corpus decreases. Future work also comprises the enhancement of the matching process by using metadata such as location, URLs or further sentiment analysis. Additionally, applying CF techniques for the exploitation of the social ties of the user are subject to future work. In order to evaluate the approach from a user's point-of-view, online user tests are also part of the future work.

[5] `http://serendip.me`

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proc. of the 20th Intl. Conf on Very Large Data Bases*, pages 487–499, 1994.

[3] L. Baltrunas and X. Amatriain. Towards Time-Dependant Recommendation based on Implicit Feedback. *Workshop on ContextAware Recommender Systems CARS 2009 in ACM Recsys*, 2009:1–5.

[4] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The Million Song Dataset. In *Proc. of the 12th Intl. Conf. on Music Information Retrieval*, 2011.

[5] Ò. Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[6] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and Tweet: Experiments on Recommending Content from Information Streams. In *Proc. of the 28th Intl. conference on Human Factors in Computing Systems*, pages 1185–1194. ACM, 2010.

[7] J. Hannon, M. Bennett, and B. Smyth. Recommending Twitter Users to Follow using Content and Collaborative Filtering Approaches. In *Proc. of the 4th ACM Conf. on Recommender Systems*, pages 199–206. ACM, 2010.

[8] M. Kaminskas and F. Ricci. Location-Adapted Music Recommendation Using Tags. In *User Modeling, Adaption and Personalization 2011, Girona, Spain, July 11-15, 2011*, volume 6787 of *LNCS*, pages 183–194. Springer, 2011.

[9] J. Lee and J. Lee. Context Awareness by Case-based Reasoning in a Music Recommendation System. In *Proc. of the 4th Intl. Conference on Ubiquitous Computing Systems*, pages 45–58. Springer, 2007.

[10] C. Mesnage, A. Rafiq, S. Dixon, and R. Brixtel. Music Discovery with Social Networks. In *Proc. of the Workshop on Music Recommendation and Discovery 2011 in conjunction with ACM RecSys*, volume 793, pages 1–6. CEUR-WS, 2011.

[11] A. Passant. dbrec - Music Recommendations Using DBpedia. *The Semantic Web–ISWC 2010*, pages 209–224, 2010.

[12] O. Phelan, K. McCarthy, and B. Smyth. Using Twitter to Recommend Real-Time Topical News. In *Proc. of the third ACM conference on Recommender systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[13] A. Schmidt, M. Beigl, and H. Gellersen. There is more to Context than Location. *Computers & Graphics*, 23(6):893–901, 1999.

[14] E. Zangerle, W. Gassler, and G. Specht. Using Tag Recommendations to Homogenize Folksonomies in Microblogging Environments. In *Social Informatics*, volume 6984 of *LNCS*, pages 113–126. Springer, 2011.