Prediction of user listening contexts for music playlists

Jeong Choi *
Knowledge AI Lab.
NCSOFT
jchoi@ncsoft.com

Anis Khlif
Deezer Research
akhlif@deezer.com

Elena V. Epure
Deezer Research
eepure@deezer.com

Abstract

In this work, we set up a novel task of playlist context prediction. From a large playlist title corpus, we manually curate a subset of multilingual labels referring to user activities (e.g. '*jogging*', '*meditation*', '*au calme*'), which we further consider in the prediction task. We explore different approaches to calculate and aggregate track-level contextual semantic embeddings in order to represent a playlist and predict the playlist context from this representation. Our baseline results show that the task can be addressed with a simple framework using information from either audio or distributional similarity of tracks in terms of track-context co-occurrences.

1 Introduction

1.1 Motivation for user listening context prediction for playlists

The origination of playlists has changed over the last two decades. Before, it used to be regarded as the work of skilled DJs or curators who had significant musical knowledge and accessibility to music databases. However, as the general music consumption has shifted to streaming services and the entire music database has became accessible to anyone, the creation of playlists has become a common way for users to organise their music catalogue in coherent collections for different listening circumstances or with different themes (Pichl et al., 2016; Dias et al., 2017).

Hence, considering how pervasive playlists are in music streaming services, being able to automatically predict their possible listening contexts could enable us to perform context-aware track recommendation for playlist continuation or to generate context-centered playlist captions.

Track-level information, such as social tags, metadata and audio content, has been widely used in research efforts seeking to unveil the general musical semantics (Levy and Sandler, 2008; Nam et al., 2018) or context-related aspects (Ibrahim et al., 2020) of single tracks. However, to our knowledge, the problem of how to deduce the music listening context for playlists by relying on signals from their track constitution has not been yet researched.

Playlist titles (Deezer)	Track-level tags (Last.fm)
soiree, rock, chill, dance, cool, **sport**, pop, electro, divers, ambiance, **party**, funk, rap, **running**, love, annee 80, **voiture**, new, calme, relax, latino, **gym**, **summer**, house, oldies, classique, **apero**, mix, slow, musique	rock, pop, alternative, indie, electronic, female vocalists, favorites, Love, dance, 00s, alternative rock, jazz, beautiful, singer-songwriter, metal chillout, male vocalists, Awesome, classic rock, soul, indie rock, Mellow, electronica, 80s, folk, british, 90s, chill, american, instrumental

Table 1: 30 most commonly used titles from Deezer playlist dataset (left) and 30 most commonly used tags from Last.fm dataset (right). The bold text ones are related to *'user-context'* category, and the others are related to *'music-context'* or *'music-content'* categories.

1.2 Playlist titles as contextual cues

The word *'context'* as employed by the recommendation system community encompasses a wide range of information such as activities, demographic information, emotional states, or weather-related information (Kaminskas and Ricci, 2012). In order to infer the user listening context, very diverse sources of data such as device logs are necessary (Cunningham et al., 2008; Wang et al., 2012; Gillhofer and Schedl, 2015), although in practical scenarios it is very challenging to access most of them while not invading user privacy.

The titles of user-created playlists, on the contrary, frequently encode information with regard to specific listening contexts and appear often as pub-

* This work was continued from the author's internship at Deezer in the Research team.

lic user information (Pichl et al., 2015). These titles are noisy since they are crowd-sourced. However, a sufficiently large corpus can give statistically meaningful cues. In this work, we utilize a large playlist dataset where each playlist has associated a user-created title, which we leverage to infer the listening context that a playlist would fit.

Unlike the track-level tags or metadata, a playlist title is more likely to represent the user listening context of the corresponding sequence of music tracks. As shown in Table 1, among the 30 most commonly used playlist titles in the Deezer playlist dataset, 8 are related to *user-context* category rather than to *music-context* or *music-content* categories (Schedl, 2013) compared to none in the track-level tags dataset (Last.fm).

While there have been multiple research works that leverage playlist titles as supplementary information for a music recommendation or playlist continuation task (Pichl et al., 2015; Zamani et al., 2019), the playlist title prediction task has not been studied. In the current work, we focus on a subset of titles referring to the context. However, the method we explore could be easily adapted to new title categories.

1.3 Context-related title prediction for playlists

The largely overlapping track-level information between different playlist titles can pose difficulties for the playlist title prediction task. For example, tracks in a playlist with the title *'running'* might be very similar to ones in a playlist with the title *'workout'* (Ibrahim et al., 2020). A previous work (Pichl et al., 2015) has tried clustering playlists with lemmatized titles to use as an additional feature for the recommendation system, while another research work (McFee and Lanckriet, 2012) has attempted to tackle this overlapping characteristic issue with a hypergraph model. However, the explicit distinction between different playlist contexts is left unclear even though those past works have helped improving the recommendation performance.

Here, we propose a framework to extract a semantic representation of a playlist as a low dimensional embedding related to its title or a specific desired concept such as the context, or user activities. To evaluate the representational power of these embeddings, we design and conduct activity-related title prediction experiments and compare the results obtained with different architectures.

2 Data preparation for user activity prediction from playlists

To set up a playlist dataset with activity labels, we first collected 2M user-created playlists from Deezer along with their titles. After a text cleaning and normalization [1] procedure, we chose 1,000 most commonly used playlist titles as our initial candidates.

A manual annotation experiment was further organised. Three music information retrieval researchers annotated each title as corresponding to a specific user activity or not. Then, 176 titles that were voted by at least two our of three annotators were selected (majority voting). Since Deezer playlist titles were multi-lingual, we further merged some cross-lingual synonyms into a single representative label, ending up with 60 activity categories (see Figure 1).

Playlists that contained less than 5 tracks were filtered out, leaving us with 156,269 playlists that had one of the 60 activity-related titles and 154,611 unique tracks included in these playlists.

3 Baseline playlist embedding models

Playlist embedding task is a many-to-one inference problem where sequential data inputs are aggregated to infer one embedding, in this case context-related. This problem is similar to the sentence embedding problem from the natural language processing field. Tracks are constitutive elements of a playlist as words are of a sentence (Kalchbrenner et al., 2014).

3.1 Using title-track matrix factorization (MF) based embeddings

Our first approach is to apply a 2-step procedure. We first compute track-level semantic embeddings based on title annotations in the playlist corpus. Then, for a given playlist, we aggregate all the track-level embeddings to make a sequence-level prediction (detailed in Section 3.3)

We aim to extract track-level embeddings that represent the 'distributional similarity' of tracks. That is, the embeddings of tracks that occur together more often (are similarly distributed) within playlists with the same title will be trained to be

[1] We lowercase and remove special characters, although we keep emoji's and some of widely used combinations of special characters manually chosen. (e.g.'<3' or ':)')

closer. This is a basic strategy to learn word embeddings and train such semantic models in the natural language processing field (Mikolov et al., 2013; Pennington et al., 2014).

By seeing a playlist as a sentence and a track as a word, we can apply any of widely used modelling techniques that extract the semantic (thematic) embedding of each track, such as Latent Dirichlet Allocation, Skip-gram (implicit matrix factorization), Word2vec, GloVe etc. (Blei et al., 2003; Levy and Goldberg, 2014; Mikolov et al., 2013; Pennington et al., 2014). Another option is to simply construct a matrix of playlist titles and track counts to conduct singular value decomposition or matrix factorization, and thus get an embedding for each track (Sarwar et al., 2001; Zhou et al., 2008; Hu et al., 2008).

Among these options, we chose the simple matrix factorization that allowed the extraction of track embeddings along with title embeddings simultaneously. We used the playlists in the training set to construct the 'title-by-track co-occurrence' matrix by adding up all track counts from playlists that are annotated with the same title. We then normalized the matrix track-wise after computing TF-IDF values. We applied alternating least square algorithm (Bell and Koren, 2007) to factorize the matrix, resulting in a 1,000-dim feature vector for each track and title.

3.2 Using audio-based embeddings

Our second approach is to learn track embeddings directly from the audio content. We set up a CNN architecture using a mel-spectrogram input that were computed with 22,050 Hz sampling rate, 1,024 FFT size, 512 hop size, and 128 mel bins. A 3-second long mel-spectrogram segment is put into the network with 5 layers of 1D convolution. The network outputs 50-dim feature vector for each segment, and we average them to end up with a 50-dim embedding for each track. In this case, track-level audio embeddings are jointly trained with the aggregated playlist embeddings in an end-to-end manner.

3.3 Aggregation techniques of track embeddings into playlist embedding

The aggregation of track embeddings into a playlist representation is done in two ways: one is by simply averaging track-level embeddings and the other is by using a single-layered LSTM network that takes a sequence of track embeddings as an input. After computing the aggregated information for a single playlist, the resulted playlist embedding is used as an input to a fully connected layer and a softmax layer that outputs the prediction for one of the 60 activity labels.

Model	MRR	FH@1	FH@5	MAP@5
MF-emb_AVG	0.531	0.360	0.753	0.508
MF-emb_LSTM	0.516	0.347	0.731	0.491
Audio_AVG	0.533	0.363	0.754	0.510
Audio_LSTM	0.540	0.369	0.761	0.517

Table 2: Baseline results on the playlist activity prediction task. *MF-emb* denotes the matrix factorization based embedding model, and *Audio* denotes the audio-based embedding model. (MRR : mean reciprocal rank / FH:flat hit / MAP:mean average precision)

4 Results and discussion

As shown in Table 2, models using audio-based embeddings performed slightly better than the ones using the MF-based embeddings. One interesting finding was that, for models using the MF-based embeddings, the model was very prone to overfit to the training set. This could be because the track-level input embeddings were computed from the matrix that partly originated from the playlist-title table that the model was being trained to predict. In this case, the simple approach of averaging track embeddings ended up performing better than making use of track-level details or the sequential information. On the other hand, for models using the audio-based embeddings, a more complex architecture that considers track-level details and the sequential order performed better, as expected.

Investigating the prediction performance on each title (see Figure 1), '*worship*', '*sunset*', '*sport*', and '*sleep*' were the most accurately predicted ones for all the models. However, we are facing a class imbalance problem where models are misguided to predict a title of the largest sample size when playlists with different titles have similar sequences of tracks. For example, for playlists labeled with '*caminhada* (walk)', '*marathon*', or '*joggin*', models would be trained to predict as '*run*' to simply achieve higher overall accuracy.

Comparing results from different input representations, '*apres ski*' and '*sex*' were more accurately predicted by the audio-based embedding models, while '*training*' was more accurately predicted by the MF embedding-based models.

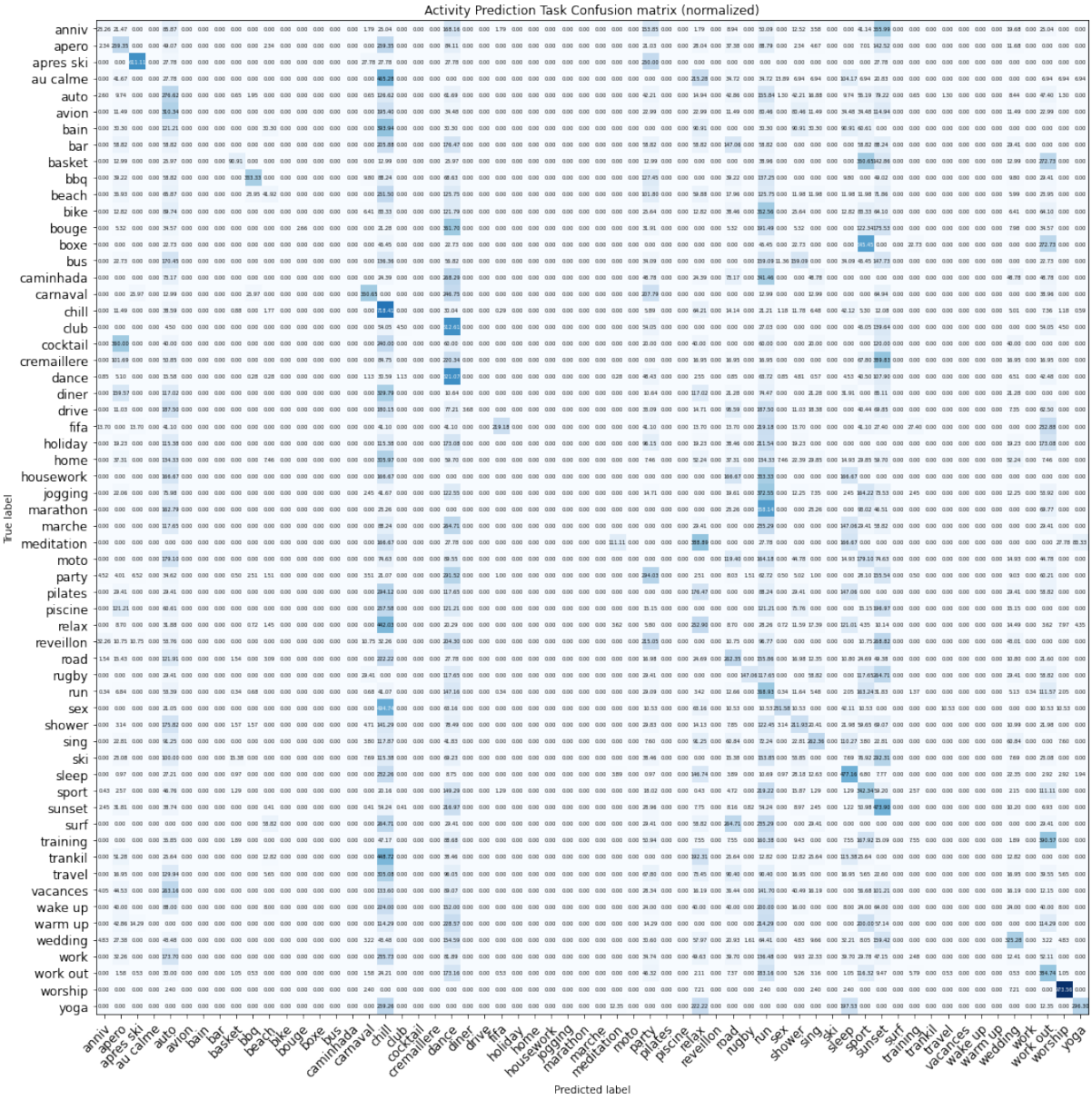

Figure 1: Confusion matrix of activity prediction results from one of the baseline models. (the audio embedding LSTM model)

Our initial results show that there is a large room to discover about how each playlists is constructed for different user listening contexts. Almost half of the activity titles could not be predicted correctly even for a single playlist. For the future work, we plan to improve the selection of the representative context titles, handle the class imbalance problem, and experiment more advanced architectures to aggregate the track-level sequential information. A multi-modal approach combining the two input representations along with any extra information such as lyrics, track metadata or user embeddings could also be promising.

References

Robert M Bell and Yehuda Koren. 2007. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 43–52. IEEE.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Stuart Cunningham, Stephen Caulder, and Vic Grout. 2008. Saturday night or fever? context-aware music playlists. *Proc. Audio Mostly*.

Ricardo Dias, Daniel Gonçalves, and Manuel J Fonseca. 2017. From manual to assisted playlist cre-

ation: a survey. *Multimedia Tools and Applications*, 76(12):14375–14403.

Michael Gillhofer and Markus Schedl. 2015. Iron maiden while jogging, debussy for dinner? In *International Conference on Multimedia Modeling*, pages 380–391. Springer.

Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee.

Karim M Ibrahim, Jimena Royo-Letelier, Elena V Epure, Geoffroy Peeters, and Gaël Richard. 2020. Audio-based auto-tagging with contextual tags for music. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20. IEEE.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Marius Kaminskas and Francesco Ricci. 2012. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2-3):89–119.

Mark Levy and Mark Sandler. 2008. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 37(2):137–150.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Brian McFee and Gert RG Lanckriet. 2012. Hypergraph models of playlist dialects. In *ISMIR*, volume 12, pages 343–348. Citeseer.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Juhan Nam, Keunwoo Choi, Jongpil Lee, Szu-Yu Chou, and Yi-Hsuan Yang. 2018. Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach. *IEEE signal processing magazine*, 36(1):41–51.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Martin Pichl, Eva Zangerle, and Günther Specht. 2015. Towards a context-aware music recommendation approach: What is hidden in the playlist name? In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1360–1365. IEEE.

Martin Pichl, Eva Zangerle, and Günther Specht. 2016. Understanding playlist creation on music streaming platforms. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 475–480. IEEE.

Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295.

Markus Schedl. 2013. Ameliorating music recommendation: Integrating music content, music context, and user context for improved music retrieval and recommendation. In *Proceedings of international conference on advances in mobile computing & multimedia*, pages 3–9.

Xinxi Wang, David Rosenblum, and Ye Wang. 2012. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 99–108.

Hamed Zamani, Markus Schedl, Paul Lamere, and Ching-Wei Chen. 2019. An analysis of approaches taken in the acm recsys challenge 2018 for automatic music playlist continuation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(5):1–21.

Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pages 337–348. Springer.