Predicting the Listening Contexts of Music Playlists Using Knowledge Graphs

Giovanni Gabbolini(✉) and Derek Bridge

Insight Centre for Data Analytics, School of Computer Science & IT,
University College Cork, Cork, Ireland
giovanni.gabbolini@insight-centre.org, d.bridge@cs.ucc.ie

Abstract. Playlists are a major way of interacting with music, as evidenced by the fact that streaming services currently host billions of playlists. In this content overload scenario, it is crucial to automatically characterise playlists, so that music can be effectively organised, accessed and retrieved. One way to characterise playlists is by their listening context. For example, one listening context is "workout", which characterises playlists suited to be listened to by users while working out. Recent work attempts to predict the listening contexts of playlists, formulating the problem as multi-label classification. However, current classifiers for listening context prediction are limited in the input data modalities that they handle, and on how they leverage the inputs for classification. As a result, they achieve only modest performance. In this work, we propose to use knowledge graphs to handle multi-modal inputs, and to effectively leverage such inputs for classification. We formulate four novel classifiers which yield approximately 10% higher performance than the state-of-the-art. Our work is a step forward in predicting the listening contexts of playlists, which could power important real-world applications, such as context-aware music recommender systems and playlist retrieval systems.

Keywords: Music playlists · Context-awareness · Recommender systems

1 Introduction

Music is commonly organised in some form of a playlist. According to a standard definition, a playlist is a sequence of music songs [5]. Playlists are a popular feature of music streaming services. Users consume playlists for 31% of their total listening time [31]; and 55% of users create their own playlists [27]. Playlists are also created for users by professional editors and by algorithms. For instance, the popular music streaming service Spotify was hosting more than four billion playlists in 2021.[1] In this content overload scenario, it is crucial to automatically characterise playlists, so that music can be effectively organised, accessed and

[1] https://backlinko.com/spotify-users.

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13980, pp. 330–345, 2023.
https://doi.org/10.1007/978-3-031-28244-7_21

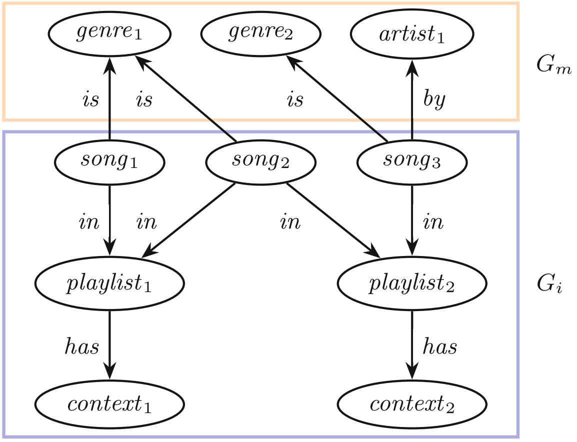

Fig. 1. A knowledge graph representing two playlists and three songs in total. The bottom and top boxes indicate two portions: G_i, which contains song, playlist and listening context nodes, and G_m, which contains metadata nodes, such as musical genres.

retrieved [9]. A common approach is playlist tagging, which is the task of assigning to a playlist one or more tags, drawn from a fixed vocabulary of tags. For example, [14] proposes a dataset of playlists annotated with a variety of different tags, like musical genres or decades. Similarly, [8] proposes a dataset of playlists annotated with listening context tags. Examples of listening context tags are "workout" and "party", which characterise playlists suited to be listened to by users while working out, and while having a party.

Listening context tags are interesting because they are user-centered, rather than music-centered [30]. For example, musical genre and decade tags refer to music. On the other hand, listening context tags refer to how people listen to music. As such, the accurate prediction of listening contexts can allow advances at the intersection of music information retrieval (MIR) and human-computer interaction (HCI), such as context-aware music recommendation [33]. In fact, recommending the right playlist at the right time is only possible if the listening context suited to listening to the playlist is known.

To the best of our knowledge, there exists only one attempt to predict the listening context of music playlists: [8]. The authors of [8] set up a multi-label classification problem, in which playlists are classified for their listening contexts, and they propose four classifiers: two matrix factorisation (MF)-based classifiers, that work by counting how many times a song is associated with each playlist listening context, and two convolutional neural network (CNN)-based classifiers, that work with song audio. However, these classifiers are limited in that they do not incorporate song metadata, such as musical genres.

In this paper, we formulate two novel knowledge graph (KG)-based classifiers. KGs are a powerful data model, suitable for storing heterogeneous information

[34]. Figure 1 depicts a KG like those we use, made up of two distinct portions: G_i and G_m. The portion G_i represents the membership of songs to playlists, and of playlists to listening contexts. The portion G_m represent song metadata, solving the limitation of existing classifiers that they do not use song metadata. The KG-based classifiers that we propose work by building a KG, such as the one depicted in Fig. 1, embedding the KG, so that each node and edge is transformed to a feature vector, and using the song embeddings to predict the listening contexts of playlists.

We benchmark the classifiers with a dataset of playlists annotated with their listening contexts, similar to the one proposed in [8]. The two KG-based classifiers we propose achieve approximately 10% higher performance than the existing predictors. A sensitivity analysis reveals that the KG-based classifiers can incorporate song metadata effectively.

However, the two KG-based classifiers do not consider song audio. So, we formulate another two novel predictors, as the hybrid of the CNN-based and KG-based classifiers. As expected, the hybrid classifiers outperform MF-based, KG-based and CNN-based predictors, setting the new state-of-the-art performance in the task.

We release the source code and the dataset that supports our work here, so as to allow reproducibility and foster new research on the subject.[2]

In summary, our contributions are:

1. The first two KG-based listening context predictors of music playlists that incorporate song metadata;
2. Another two novel predictors that incorporate KGs and song audio;
3. A comparison of the predictors reporting approximately 10% higher performance than the state-of-the-art, and showing the impact of song metadata on performance.

The rest of the paper is organised as follows: in Sect. 2, we review related work on music listening contexts, and especially work that looks into how music consumption changes in different listening contexts. We also review related work in music tagging. In Sect. 3, we describe our four novel classifiers for predicting the listening context of music playlists. In Sect. 3, we present extensive experiments that compare the novel classifiers to existing classifiers, and validate the design of the novel classifiers with a sensitivity analysis. Section 5 concludes the paper and outline future work.

2 Related Work

The task of tagging can be defined as marking content with descriptive terms, also called keywords or tags, drawn from a fixed vocabulary [16]. Content can refer to different objects, such as text, audio, images or video. For example, [7] propose an approach for tagging an image with its objects, such as: "fish",

[2] https://github.com/GiovanniGabbolini/playlist-context-prediction.

"plane" or "shoe". And, [25] survey tagging systems in the text, image and music domains.

In this work, we focus on the music domain, as tagging is a major topic in music information retrieval (MIR). Music tagging is the task of classifying music in one or more tag classes. As such, the vocabulary of tags is typically assumed to be fixed. One common setup is song tagging, where single songs are classified. [36], for example, offers a comparison of recent Convolutional Neural Network (CNN)-based classifiers: a CNN extracts learned features from the audio of a song, and leverages these features to output appropriate tags. Similarly, the state-of-the-art classifiers proposed in [10,29,35] are CNN-based. Progress in song tagging is enabled by the availability of large scale datasets, such as the Million Songs Dataset [3], the MagnaTagATune Dataset [24] and the MG-Jamendo Dataset [4]. These datasets contain songs annotated with tags of several categories: genre tags (*e.g.* "jazz"), instrumentation tags (*e.g.* "guitar"), decade tags (*e.g.* "80s"), mood tags (*e.g.* "happy") and listening context tags (*e.g.* "party"). A related (but different) task to song tagging is playlist tagging, where a list of songs is tagged, instead of a single song. [14] proposes a dataset of playlists annotated with a variety of different tags, like genre tags or decade tags. Classifiers for song tagging can be extended to do playlist tagging. For example, [8] proposes a CNN-based playlist classifier, with an architecture similar to the CNN-based song classifiers.

Previous work shows that music listening behaviour depends on the listening context [11,17]. For example, users listen to one type of music while having a party, to another type of music while spending time alone, and to another type while working. Context-aware music recommender systems [33] address the user's need to access the right music in the right context. Applications include: context-aware song and playlist recommendation, and context-aware playlist continuation [31]. Predicting the listening context that suits some music is a first step towards context-awareness. Hence, some of the recent work on music tagging focuses on listening context tags only. For example, [19,20] propose a dataset of songs annotated with listening context tags, and a baseline CNN-based classifier. And, [8] proposes a dataset of playlists annotated with listening context tags, such as "workout" and "party", and four baseline classifiers: two CNN-based and another two MF-based classifiers.

Our work here is on playlist tagging, as we focus on predicting the listening contexts of playlists. We build on [8], as we propose four novel classifiers, which outperform the four classifiers that they propose, setting the new state-of-the-art performance in the task.

3 Method

Predicting the listening contexts of playlists is framed by the authors of [8] as a multi-label classification problem. The same authors propose four such classifiers (MF-avg, MF-seq, CNN-avg and CNN-seq). Here, we propose another four such classifiers (KG-avg, KG-seq, Hybrid-avg, Hybrid-seq). As we will explain below, six of the classifiers that we consider follow the schema depicted in Fig. 2. The two hybrid classifiers follow the schema depicted in Fig. 3. In the rest of this section, we summarise the four classifiers that were proposed in [8], and we describe the four classifiers that we propose.

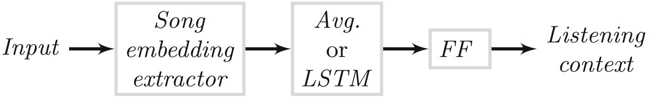

Fig. 2. Schematic architecture of MF-AVG, MF-SEQ, CNN-AVG, CNN-SEQ, KG-AVG and KG-SEQ.

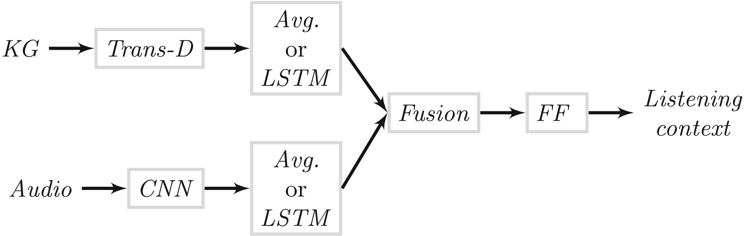

Fig. 3. Schematic architecture of HYBRID-AVG and HYBRID-SEQ.

3.1 Matrix Factorisation-Based

The two matrix factorisation (MF)-based classifiers (MF-AVG and MF-SEQ), originally proposed in [8], take as input a matrix $\mathbf{X} \in \mathbb{R}^{N,M}$ where N is the number of songs and M is the number of listening contexts. The element at row n and column m of $\mathbf{X}$ is equal to the number of times the n^{th} song appears in playlists that have the m^{th} context. The matrix $\mathbf{X}$ is factorised into two matrices, $\mathbf{S} \in \mathbb{R}^{N,H}$ and $\mathbf{C} \in \mathbb{R}^{H,M}$, using WR-MF, which is the MF procedure described in [18], so that $\mathbf{SC} \approx \mathbf{X}$. H is the embedding dimension, which is a hyper-parameter of WR-MF. The rows of $\mathbf{S}$ and the columns of $\mathbf{C}$ contain, respectively, song and listening context embeddings. Then, the song embedding vectors for the songs in a given playlist (a subset of the embeddings contained in $\mathbf{S}$) are either averaged song-wise (in MF-AVG) or input to a single-layered LSTM network (in MF-SEQ), to get a playlist embedding vector, which is fed into a single-layered feed-forward (FF) network that outputs a score for each listening context.

The architecture of MF-AVG and MF-SEQ fit into the schema of Fig. 2 as the matrix $\mathbf{X}$ is the *input*, and WR-MF is the *song embedding extractor*. Notice that MF-AVG and MF-SEQ work in two steps, that is the *song embedding extractor* is trained separately from the rest.

3.2 Convolutional Neural Network-Based

The two convolutional neural network (CNN)-based classifiers (CNN-AVG and CNN-SEQ), originally proposed in [8], extend the state-of-the-art in song tagging to playlist tagging. Given a song, they consider the full audio, and compute mel-spectrograms for every contiguous 3-seconds of audio. The mel-spectrogram is a hand-crafted feature extracted from audio, commonly used in many music information retrieval (MIR) tasks, such as song tagging, e.g. [10,29,35]. The mel-spectrograms are input to a Convolutional Neural Network (CNN) with five *1D*-convolutional layers, which outputs an embedding vector for every 3-seconds of audio. Such embeddings are averaged point-wise, to get one song embedding vector. Given a playlist, the song embedding vectors are computed as above, and either averaged song-wise (in CNN-AVG) or input to a single-layered LSTM network (in CNN-SEQ), to get a playlist embedding vector, which is fed into a single-layered (FF) network that outputs a score for each listening context.

The architecture of CNN-AVG and CNN-SEQ fit into the schema of Fig. 2 as the mel-spectrograms are the *input*, and the CNN is the *song embedding extractor*. Notice, however, that CNN-AVG and CNN-SEQ are end-to-end, that is the *song embedding extractor* is trained jointly with the rest.

3.3 Knowledge Graph-Based

A knowledge graph (KG) is a set of triples $G = \{(e, r, e') \,|\, e, e' \in E, r \in R\}$, where E and R denote, respectively, the sets of entities (nodes) and relationships (edges). KGs are suitable for representing heterogeneous information [34]. For example, [28] builds a KG representing users, their interactions with songs, and acoustical metadata, such as what musical instruments are played in the songs.

The information we want to represent is: songs; playlists; listening contexts; and song metadata. So, we build a KG composed of two portions. (1) G_i: the portion containing song nodes, playlist nodes and listening context nodes. These nodes are connected by edges according to membership: a song node is connected to the playlist nodes the song belongs to, and a playlist node is connected to its listening context node. (2) G_m: the portion containing song metadata, *i.e.* the record label associated with the song, the musical genres associated with the song, the year and the month when the song was released, the artist of the song, the city and the country where the artist is currently based, and where they were born. We selected these items of metadata empirically, through informal experimentation, and by taking inspiration from previous work; for example, [21] finds that the release year of a song can be a predictor for the listening context. In future work, G_m can be readily expanded to include more song metadata, such as information extracted from song lyrics. For each piece of song metadata, there is a node in G_m. Song nodes are connected by edges to their metadata nodes. Song metadata may be missing, *e.g.* we may not know the record label for a particular song. We obtain metadata from the crowd-sourced database MusicBrainz.[3]

Figure 1 depicts a KG, like those that we build.

We embed the KG using the TRANS-D algorithm, which is a state-of-the-art KG embedding algorithm [12]. TRANS-D produces an embedding vector for every node and edge in the KG, in such a way that the topology of the KG is preserved. In particular, given a KG G, and given a triple $(e, r, e') \in G$, TRANS-D produces three embedding vectors v_e, v_r and $v_{e'}$ that satisfy a relationship similar to $v_e + v_r \approx v_{e'}$, for every triple in G. The embedding vectors of the song nodes in the KG for the songs in a playlist are either averaged song-wise (in KG-AVG) or input to a single-layered LSTM network (in KG-SEQ), to get a playlist embedding vector, which is fed into a (FF) network, that outputs a score for each listening context.

The architecture of KG-AVG and KG-SEQ fit into the schema of Fig. 2 as the KG we build is the *input*, and TRANS-D is the *song embedding extractor*. Notice that KG-AVG and KG-SEQ work in two steps, that is the *song embedding extractor* is trained separately from the rest.

The MF-based and KG-based algorithms both leverage information about listening contexts when computing song embeddings. However, KG-based algorithms exploit that information more effectively. For example, let us consider the scenario depicted by the portion G_i of the KG in Fig. 1 where there are two playlists, $playlist_1$ and $playlist_2$, whose listening contexts are respectively $context_1$ and $context_2$, and which contain respectively the songs $song_1$ & $song_2$ and $song_2$ & $song_3$. MF song embeddings are aligned with their listening contexts, as explained in Sect. 3.1. In the example above, the MF embedding of $song_1$ is aligned with $context_1$, the MF embedding of $song_3$ is aligned with $context_2$, and the embedding of $song_2$ is aligned with both $context_1$ and $context_2$. However, $song_1$ and $song_2$ are in the same playlist ($playlist_1$). As such, we expect the embedding of $song_1$ to be aligned, to some extent, also with $context_2$, and not only with $context_1$; similarly for the embedding of $song_3$. That is, MF-based algorithms 'short-circuit' the representation of playlists by modelling the association of songs to playlist listening contexts directly. KG embeddings preserve the topology of the KG, and so can overcome the short-circuiting problem of the MF-algorithms. In the example above, the songs in G_i are all connected with each other, via the explicit representation of the playlists as well as the listening contexts. That is, the embeddings of $song_1$, $song_2$ and $song_3$ are all aligned, to some extent with $context_1$, and to some other extent with $context_2$. The short-circuiting problem undermines the performance of the MF-based classifiers, as we empirically prove in Sect. 4.3. In a similar vein, [26] propagates tags among songs in the same playlists, and measure an increase in performance.

[3] https://musicbrainz.org.

3.4 Hybrid

The CNN-based classifiers and the KG-based classifiers differ on their input data, as the CNN-based classifiers rely on song audio, while the KG-based classifiers rely on a KG representation of songs, playlists, listening contexts, and song metadata. The audio and the KG differ in modality, as well as availability. For example, while song audio is available for every song in the catalogue, a KG such as the one we use may represent the most famous songs well, but it may fail to represent properly more niche songs, which is a manifestation of the long-tail problem [22], and it may also fail to represent newly-released songs. To address this limitation, we complement the KG-based classifiers with the CNN-based classifiers, by formulating two hybrid classifiers.

The hybrids work by jointly running a KG-based classifier (KG-AVG or KG-SEQ) and a CNN-based classifier (CNN-AVG or CNN-SEQ), and by fusing the two playlist embedding vectors that they compute, before they are passed to a single-layered FF network that outputs a score for each listening context. We refer to HYBRID-AVG as the hybrid of KG-AVG & CNN-AVG and to HYBRID-SEQ as the hybrid of KG-SEQ & CNN-SEQ. The architecture of the two hybrids follow the schema of Fig. 3.

For the embedding fusion, both the audio and KG-based playlist embedding vectors are input to two separate linear layers, two separate non-linearities, and then summed point-wise, as suggested by [2]. We did experiment with other simple fusion strategies, *e.g.* concatenation, but they achieved lower performance.

3.5 Implementation Details

Our implementation of CNN-AVG and CNN-SEQ is a little different from the original paper [8] as we make two simplifications. First, we use Spotify's 30-second audio previews of the songs instead of their full audio. These audio previews are freely available, unlike the full audio, which is expensive to access due to copyright restrictions. Moreover, the usage of audio previews make our work reproducible. Second, we average the 3-second mel-spectrograms of a song point-wise in input to the CNN. As such, the CNN receives only one spectrogram, and outputs the song embedding directly. This second simplification saves computing resources. In Sect. 4, we show that our implementation of the CNN-based models outperforms the MF-based models, which is consistent with the original paper. More specifically, our implementation of CNN-SEQ achieves 7% higher performance than MF-SEQ, which is consistent with the original paper; similarly for CNN-AVG and MF-AVG. Given those results, we are confident that our implementations of the CNN-based models, although simplified, are as valid as the original implementations presented in [8].

We compute the mel-spectrograms for the CNN-AVG and CNN-SEQ classifiers with 22,050 Hz sampling rate, 1,024 FFT size, 512 hop size, and 128 mel bins. We set hyper-parameters of the MF and CNN-based classifiers as in the original paper [8]. That is, we set the song embedding dimension to 50, and we use *ReLU* as the non-linearity. We do the same in the KG-based and hybrid classifiers. We train the classifiers with early-stopping, monitoring *FH@1* on the validate set, with patience equal to ten. We tune other hyper-parameters of the eight classifiers (learning rate, weight decay and batch size) using bayesian optimisation [32]. We fix the number of trials of the bayesian optimiser to 20. For the WR-MF and Trans-D embedding procedures, we use the default parameters and we set the number of epochs to ensure convergence of the loss function.

For other implementation details, we refer the reader to the source code that supports our work here.

4 Experiments

We compare the classifiers described in Sect. 3, and variants of those, on their performance in predicting the listening context of music playlists.

4.1 Dataset

We use a dataset of playlists annotated with their listening contexts. The dataset was annotated by the authors of [8], starting from user playlists contained in the Spotify Million Playlist Dataset (MPD) [6], and retaining only the portion of playlists that have a listening context as title.[4] Examples of listening contexts present in the dataset are: driving, studying and summertime. For other examples, we refer the reader to the dataset that supports our work here. Also, we refer the reader to [8] for more information on the annotation procedure. Each playlist is annotated with one listening context. We split the dataset randomly into train, validate and test sets, accounting respectively for 60%, 20% and 20% of the total playlists. Similar to [8], we filter out songs that occur in the validate and test sets but not in the train set, as some classifiers cannot handle at testing time songs not seen at training time. The classifiers that have this limitation are MF-avg, MF-seq, KG-avg and KG-seq. They work by training a *song embedding extractor* model in a first step, separately from the classifier that outputs the listening context, see Sects. 3.1 and 3.3. As a result, embeddings for songs not present at training time are not available at test time. In a real world scenario, where new releases are frequently added to the songs catalogue, it would be necessary to incrementally train the models so that the training set covers all songs in the catalogue. An alternative is to use CNN-avg and CNN-seq, as they rely on the audio signal, which is available for songs not seen at training time.

Table 1 contains statistics of the dataset that we use (train, validate and test splits together).

[4] The dataset we use is not the one used in [8], which is proprietary, but it was supplied to us by the authors of [8] as a dataset annotated with the same procedure, and in which similar results can be obtained.

Table 1. Dataset statistics.

Statistic	Value
Number of playlists	114,689
Average playlist length	62.6
Number of unique songs	418,767
Number of unique listening contexts	102

4.2 Metrics

We call D the test set, and we call p a playlist in the test set, that is $p \in D$. We call $|D|$ the number of playlists in the test set. The classifiers described in Sect. 3 predict a score for each listening context. As such, given a playlist, a classifier predicts a ranking of listening contexts, by decreasing score. Given a ranking of listening contexts for a playlist p, we call $rank_p$ the position of the correct listening context in the ranking. For example, if a classifier assigns the highest score to the correct listening context, then $rank_p = 1$. Instead, if the classifier assigns the lowest score to the correct listening context, then $rank_p = 102$ (see Table 1).

We compare the classifiers for their performance in predicting the listening contexts of the playlists in D. We measure performance with four metrics, as in [8]:

Flat hits (*FH@1*, *FH@5*). Flat hits is the percentage of playlists D such that $rank_p \leq k$. In our case, since the goal is classification rather than retrieval, we consider only $k = 1$ and $k = 5$ and no higher values for k. In formulas:

$$FH@k = \frac{1}{|D|} \sum_{p \in D} \mathbb{1}(rank_p \leq k)$$

where $\mathbb{1}(rank_p \leq k)$ is the indicator function. That is, $\mathbb{1}(rank_p \leq k) = 1$ if $rank_p \leq k$ and 0 otherwise. In other words, *FH@1* is the percentage of playlists for which the classifier predicts the listening context correctly. And, *FH@5* is the percentage of playlists for which the classifier predicts the correct listening context among the first five predictions.

Mean reciprocal rank (*MRR*). The reciprocal rank is the reciprocal of $rank_p$. The *MRR* is the average of those reciprocals ranks. In formulas:

$$MRR = \frac{1}{|D|} \sum_{p \in D} \frac{1}{rank_p}.$$

Mean average precision (*MAP@5*). *MAP* is equivalent to *MRR*, except that we set the reciprocal rank to 0 when $rank_p > k$[5]. That is, if $rank_p > k$ for

[5] Our formulation of MAP is different from others, which allow for multiple relevant items. In our case, there is only one relevant item: the correct listening context.

Table 2. Performance of the classifiers.

	FH@1	*FH@5*	*MRR*	*MAP@5*
MF-avg	0.299	0.536	0.416	0.386
MF-seq	0.327	0.595	0.452	0.423
CNN-avg	0.291	0.583	0.425	0.395
CNN-seq	0.352	0.639	0.484	0.456
KG-avg	0.388	0.678	0.521	0.495
KG-seq	0.389	0.678	0.520	0.494
Hybrid-avg	**0.395**	**0.687**	**0.528**	**0.503**
Hybrid-seq	0.389	0.678	0.520	0.495

every $p \in D$, then $MAP@k = 0$. In our case, we consider $k = 5$. In formulas:

$$MAP@k = \frac{1}{|D|} \sum_{p \in D} \frac{1}{rank_p} \times \mathbb{1}(rank_p \leq k).$$

On the one hand, *FH@k* disregards the actual position of the correct listening context in the ranking, but counts how frequently this position is lower than a threshold k. On the other hand, *MAP@k* and *MRR* do account for the actual position of the correct listening context in the ranking. Therefore, these metrics give a multi-sided view of the classifiers' performance.

We set up significance tests to check whether differences in performance are statistically significant or not. Following [13], we set up a t-test for *MRR* and *MAP@5*, and a paired bootstrap test for *FH@1* and *FH@5*. Similar to [23], we fix the number of bootstrap replicas to 1000.

4.3 Results

We conduct two experiments: a comparison with the state-of-the-art, and a sensitivity analysis.

Comparison with State-of-the-Art. We measure the performance of the classifiers that we propose (KG-avg, KG-seq, Hybrid-avg, Hybrid-seq), and the performance of the state-of-the-art baselines, *i.e.* the existing listening context classifiers (MF-avg, MF-seq, CNN-avg, CNN-seq). The results are in Table 2.

The classifiers that we propose outperform the baselines by a considerable amount. Hybrid-avg scores highest performance, improving by approximately 10% over the baselines. The improvement in performance is statistically significant ($p < 10^{-4}$). In general, all the classifiers we propose improve performance over the baselines ($p < 10^{-4}$).

The improvement in performance has real world relevance. For example, Hybrid-avg achieves 12% higher *FH@1* than the best baseline (0.395 *vs* 0.352),

Table 3. Performance of KG-based classifiers with (w) and without (wo) song metadata.

	FH@1	*FH@5*	*MRR*	*MAP@5*
KG-AVG wo metadata	0.375	0.665	0.507	0.481
KG-AVG w metadata	0.388	0.679	0.521	0.495
KG-SEQ wo metadata	0.382	0.668	0.513	0.487
KG-SEQ w metadata	0.388	0.679	0.520	0.495

which means than in a sample of 1000 playlists, our algorithm predicts the listening context correctly 395 *vs* 352 times, on average. Considering that the current databases contain millions of playlists, the 12% increase over the baselines is particularly 'tangible'.

We notice that the more complex SEQ variants of the algorithms are not always superior to their simpler AVG variant. MF-SEQ and CNN-SEQ have higher performance than, respectively, MF-AVG and CNN-AVG ($p < 10^{-4}$). But we do not find any statistically significant differences between the performance of KG-AVG and KG-SEQ, while HYBRID-SEQ has lower performance than HYBRID-AVG ($p < 10^{-4}$). Probably, the architecture of HYBRID-SEQ is too complex for the task at hand, and may overfit the training set, while the simpler HYBRID-AVG generalises better to new data. Moreover, the result corroborates previous work [8], where the SEQ variant is found to be sometimes superior and sometimes inferior to the AVG variant.

The hybrid classifiers are the combination of the (audio) CNN-based and KG-based classifiers. Accordingly, HYBRID-AVG has higher performance than CNN-AVG and KG-AVG. Though statistically significant ($p < 10^{-4}$), the increase in performance is only slight. We can understand the result by looking at the literature on the well-studied task of music similarity [1]. Flexer [15] shows that increasing the performance of similarity algorithms is particularly challenging after a certain threshold, as there exists an upper bound to performance, caused by the low agreement of different users in the perception of music similarity. Likewise, humans can have different perceptions of the right listening context for a given playlist. In the dataset we use, each song is associated with 17 different playlist listening contexts, on average. As such, we expect that increasing the performance of classifiers can become particularly challenging after a certain threshold. For example, HYBRID-SEQ has higher performance than CNN-SEQ ($p < 10^{-4}$), but not over KG-SEQ (no statistically significant difference).

Sensitivity Analysis. KG-based classifiers have as input a KG with songs, playlists, their listening contexts (portion G_i) and song metadata (portion G_m). We measure the performance of variants of the KG-based classifiers that have as input only the portion G_i of the full KG. The results are in Table 3, and show an increase in performance when using metadata ($p < 10^{-4}$). This indicates that the KG-based classifiers make effective use of song metadata for predicting listening contexts. However, the increase in performance is only slight, and again can be explained by the work of Flexer [15], as in Sect. 4.3.

The G_i portion of the KG contains the same information as the input to the MF-based classifiers, *i.e.* playlist listening contexts. However, as argued in Sect. 3.3, MF-based classifiers suffer from what we called the playlist short-circuiting problem, *i.e.* they model the association of songs to playlist listening contexts directly, while KG-based classifiers do not. A comparison of the results of the KG-based classifiers without metadata in Table 3 and the MF-based classifiers in Table 2 reveals the consequences of these two ways of modelling the information. The comparison shows that the KG-based algorithms exploit that information more effectively, since their results are significantly superior to those of the MF-based algorithms ($p < 10^{-4}$).

5 Conclusions and Future Work

We propose four novel systems for predicting the listening contexts of music playlists, which include, for the first time, song metadata in their models. In two of them, we represent songs, playlists, listening contexts and song metadata in a KG, that we embed, and we use the song embeddings to make predictions. In the other two, we combine the KG and song audio in a unique hybrid model. We benchmark the performance of the predictors we propose, reporting an increase in performance of approximately 10% over the state-of-the-art. We also show, through a sensitivity analysis, that the KG-based predictors can incorporate the song metadata effectively. We argued that the improvement in performance that we have achieved has real world relevance.

Our work can power a number of real applications that make use of listening contexts, such as context-aware recommender systems. More generally, our work introduces a way to use KGs for effective music classification, which is an under-explored direction.

Future work include the construction of a novel playlist extender *i.e.* one that recommends songs to add to a playlist but that ensures that the new songs are suited to the playlist listening context.

Acknowledgements. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. We are grateful to Elena Epure & Romain Hennequin from Deezer, for sharing the dataset we used in the experiments, and for assisting us in the process of replicating the baselines. We are also grateful to Jeong Choi from NAVER, for assisting us in the process of replicating the baselines.

References

1. Aucouturier, J.J., Pachet, F., et al.: Music similarity measures: What's the use? In: 3rd International Society for Music Information Retrieval Conference, pp. 13–17 (2002)
2. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: a survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 423–443 (2018)
3. Bertin-Mahieux, T., Ellis, D.P., LabROSA, E., Whitman, B., Lamere, P.: The million song dataset. In: Proceedings of the 12th International Conference on Music Information Retrieval (2011)
4. Bogdanov, D., Won, M., Tovstogan, P., Porter, A., Serra, X.: The MTG-Jamendo dataset for automatic music tagging. In: Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019). Long Beach, CA, United States (2019). http://hdl.handle.net/10230/42015
5. Bonnin, G., Jannach, D.: Automated generation of music playlists: survey and experiments. ACM Comput. Surv. (CSUR) **47**(2), 1–35 (2014)
6. Chen, C.W., Lamere, P., Schedl, M., Zamani, H.: RecSys challenge 2018: Automatic music playlist continuation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 527–528. RecSys 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3240323.3240342
7. Chen, M., Zheng, A., Weinberger, K.: Fast image tagging. In: International Conference on Machine Learning, pp. 1274–1282. PMLR (2013)
8. Choi, J., Khlif, A., Epure, E.: Prediction of user listening contexts for music playlists. In: Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA), pp. 23–27 (2020)
9. Choi, K., Fazekas, G., McFee, B., Cho, K., Sandler, M.: Towards music captioning: generating music playlist descriptions. arXiv preprint arXiv:1608.04868 (2016)
10. Choi, K., Fazekas, G., Sandler, M., Cho, K.: Convolutional recurrent neural networks for music classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2392–2396. IEEE (2017)
11. Cunningham, S.J., Bainbridge, D., Falconer, A.: 'More of an art than a science': supporting the creation of playlists and mixes. In: 7th International Society for Music Information Retrieval Conference (2006)
12. Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: approaches, applications and benchmarks. Electronics **9**(5), 750 (2020)
13. Dror, R., Baumer, G., Shlomov, S., Reichart, R.: The hitchhiker's guide to testing statistical significance in natural language processing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1383–1392. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1128. https://aclanthology.org/P18-1128
14. Ferraro, A., et al.: Melon playlist dataset: a public dataset for audio-based playlist generation and music tagging. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 536–540. IEEE (2021)
15. Flexer, A.: On inter-rater agreement in audio music similarity. In: Proceedings of the 15th International Conference on Music Information Retrieval, pp. 245–250 (2014)
16. Golder, S.A., Huberman, B.A.: The structure of collaborative tagging systems. J. Inf. Sci. **32**(2), 0508082 (2006)

17. Greasley, A.E., Lamont, A.: Exploring engagement with music in everyday life using experience sampling methodology. Music Sci. **15**(1), 45–71 (2011)
18. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 263–272. IEEE (2008)
19. Ibrahim, K., Epure, E., Peeters, G., Richard, G.: Should we consider the users in contextual music auto-tagging models? In: 21st International Society for Music Information Retrieval Conference (2020)
20. Ibrahim, K.M., Royo-Letelier, J., Epure, E.V., Peeters, G., Richard, G.: Audio-based auto-tagging with contextual tags for music. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 16–20. IEEE (2020)
21. Kamehkhosh, I., Bonnin, G., Jannach, D.: Effects of recommendations on the playlist creation behavior of users. User Model. User-Adap. Inter. **30**(2), 285–322 (2020)
22. Knees, P., Schedl, M.: Contextual music meta-data: comparison and sources. In: Music Similarity and Retrieval. TIRS, vol. 36, pp. 107–132. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49722-7_5
23. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 388–395. Association for Computational Linguistics, Barcelona, Spain (2004). https://aclanthology.org/W04-3250
24. Law, E., West, K., Mandel, M.I., Bay, M., Downie, J.S.: Evaluation of algorithms using games: The case of music tagging. In: 10th International Society for Music Information Retrieval Conference, pp. 387–392 (2009)
25. Lee, S., Masoud, M., Balaji, J., Belkasim, S., Sunderraman, R., Moon, S.J.: A survey of tag-based information retrieval. Int. J. Multimedia Inf. Retrieval **6**(2), 99–113 (2017)
26. Lin, Y.H., Chung, C.H., Chen, H.H.: Playlist-based tag propagation for improving music auto-tagging. In: 2018 26th European Signal Processing Conference (EUSIPCO), pp. 2270–2274. IEEE (2018)
27. Muligan, M.: Announcing MIDiA's state of the streaming nation 2 report. https://midiaresearch.com/blog/announcing-midias-state-of-the-streaming-nation-2-report (2017). Accessed 15 Mar 2022
28. Oramas, S., Ostuni, V.C., Noia, T.D., Serra, X., Sciascio, E.D.: Sound and music recommendation with knowledge graphs. ACM Trans. Intell. Syst. Technol. (TIST) **8**(2), 1–21 (2016)
29. Pons, J., Nieto, O., Prockup, M., Schmidt, E., Ehmann, A., Serra, X.: End-to-end learning for music audio tagging at scale. In: Proceedings of the 19th International Conference on Music Information Retrieval (2018)
30. Schedl, M., Flexer, A., Urbano, J.: The neglected user in music information retrieval research. J. Intell. Inf. Syst. **41**(3), 523–539 (2013)
31. Schedl, M., Zamani, H., Chen, C.-W., Deldjoo, Y., Elahi, M.: Current challenges and visions in music recommender systems research. Int. J. Multimedia Inf. Retrieval **7**(2), 95–116 (2018). https://doi.org/10.1007/s13735-018-0154-2
32. Victoria, A.H., Maragatham, G.: Automatic tuning of hyperparameters using Bayesian optimization. Evol. Syst. **12**(1), 217–223 (2021)
33. Wang, X., Rosenblum, D., Wang, Y.: Context-aware mobile music recommendation for daily activities. In: Proceedings of the 20th ACM International Conference on Multimedia, pp. 99–108 (2012)

34. Wilcke, X., Bloem, P., De Boer, V.: The knowledge graph as the default data model for learning on heterogeneous knowledge. Data Sci. **1**(1–2), 39–57 (2017)
35. Won, M., Chun, S., Nieto, O., Serra, X.: Data-driven harmonic filters for audio representation learning. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 536–540. IEEE (2020)
36. Won, M., Ferraro, A., Bogdanov, D., Serra, X.: Evaluation of CNN-based automatic music tagging models. arXiv preprint arXiv:2006.00751 (2020)