# SEMANTICALLY MEANINGFUL ATTRIBUTES FROM CO-LISTEN EMBEDDINGS FOR PLAYLIST EXPLORATION AND EXPANSION

Ayush Patwari, Nicholas Kong, Jun Wang, Ullas Gargi YouTube Music

{patwaria,kongn,juwanng,ullas}@google.com

Michele Covell, Aren Jansen Google Research

{covell,arenjansen}@google.com

# ABSTRACT

Audio embeddings of musical similarity are often used for music recommendations and autoplay discovery. These embeddings are typically learned using co-listen data to train a deep neural network, to provide consistent tripletloss distances. Instead of directly using these co-listenbased embeddings, we explore making recommendations based on a second, smaller embedding space of humanintelligible musical attributes. To do this, we use the colisten-based audio embeddings as inputs to small attribute classifiers, trained on a small hand-labeled dataset. These classifiers map from the original embedding space to a new interpretable attribute coordinate system that provides a more useful distance measure for downstream applications. The attributes and attribute embeddings allow us to provide a search interface and more intelligible recommendations for music curators. We examine the relative performance of these two embedding spaces (the co-listen-audio embedding and the attribute embedding) for the mathematical separation of thematic playlists. We also report on the usefulness of recommendations from the attributeembedding space to human curators for automatically extending thematic playlists.

# 1. INTRODUCTION

Automatically annotating music with semantically meaningful and musically relevant attributes is an important effort with a long history [1–5]. It has become especially important as music-streaming services have made large catalogs of recorded music available to people worldwide and as the user interface to these services continues to shift towards voice activation rather than text search or graphical browsing. Describing music using such musical attributes has many applications such as:

 allowing consumers to search for music that satisfies musical, emotional or psychological constraints, using text or voice queries;

- browsing for such music using common usage patterns reflecting activities (*e.g.*, "running") or moods (*e.g.*, "chill");
- sequencing playlists for users that allows them to choose which aspect of musical similarity to maintain, rather than simply following general co-listen patterns;
- providing power users and curators the ability to program specific experiences using higher-order operators.

In this paper, we describe a system for understanding and describing music content. The paper is divided into two main parts. The first part (Section 2) describes how we extract semantically meaningful attributes from features primarily based on audio-spectrogram embeddings trained on co-listen user behavior. The second part (Section 3) explores using these semantic-attribute embeddings both to characterize and to extend professionally curated playlists.

# 2. EXTRACTING SEMANTICALLY MEANINGFUL ATTRIBUTES FROM CO-LISTEN EMBEDDINGS

We collaborated with the YouTube Music curation team to establish a prioritized list of semantically meaningful audio attributes. Several attributes are subjective, making it difficult to get enough ground truth data to support training. Typical deep networks require many labeled examples, due to the millions of trainable parameters in modern deep networks: for example, ResNet-18 (used by [6]) has around 11 million trainable parameters. We could use existing meta-data for some attributes, such as genre, but other important attributes, such as vocalness (presence of vocals) and energy, are not as prevalent. Even with genres, there is not a consistent set of labels available across different music distributors.

We first review the work of [6]: building on this work allows us to have shallow yet powerful attribute embeddings. In Subsection 2.2, we describe our approach to extracting full-track-level attributes from those co-listen-based audio embeddings. We then discuss our work in measuring attribute *consistency* across the duration of the track (Subsection 2.3). Finally, in Subsection 2.4, we report our accuracy.

<sup>©</sup> Or Ayush Patwari, Nicholas Kong, Jun Wang, Ullas Gargi, Michele Covell, Aren Jansen. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ayush Patwari, Nicholas Kong, Jun Wang, Ullas Gargi, Michele Covell, Aren Jansen, "Semantically Meaningful Attributes from Co-Listen Embeddings for Playlist Exploration and Expansion", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.*, Montréal, Canada, 2020.

# 2.1 Co-listen-based Audio Embeddings

To allow us to train only comparatively shallow networks, we build on the audio-embedding work done by [6]. In this subsection, we review the approach taken in that previous work.

In [6], an initial audio embedding was obtained using triplet loss from aggregated listening sessions. With the triplet loss, tracks that were (in aggregate) listened to together were trained to be closer in the embedding space than those that were not. The unaveraged embedding is generated using a modified version of Resnet-18, operating on overlapping 3-second audio-spectrogram windows (with a 1-second overlap). For training, [6] used random 3-second samples from the anchor, positive-example, and negative-example tracks. They evaluated by holding back 10% of the 10.5-million audio tracks in their co-listen dataset. When testing on this hold-out set, they achieve over 50% improvement in performance as [2] under the same training regime (0.079 average precision vs. 0.055 for [2]).

In this paper, we use averages of the [6] embeddings as our audio embeddings, with averaging either over the full song duration (Subsection 2.2) or over 10-second tiles (Subsection 2.3).

# 2.2 Co-Listen Embeddings to Full-Track Attributes

In this paper, we consider the following attributes:

- Genre: A subset of the full international set of genres, including only those deemed important for the US market<sup>1</sup>, specifically: Hip-Hop/Rap, R&B/Soul, Blues, Country, Jazz, Rock, Metal, Pop, Dance/Electronic, Alternative/Indie, Latin Urbano, Regional Mexican, Reggae, K-Pop, Korean Ballads, and Classical;
- Valence (or hedonic tone): a measure of the emotional positivity or negativity of the music;
- Vocalness: a measure of the prominence of speaking and singing (or even wordless screaming or humming);
- Energy: a qualitative measure of the intensity (or autonomic arousal) of the music;
- Temporal consistency of energy across a track (Subsection 2.3).

The primary source of ground-truth labels for our attribute-classification models is the team of music experts at YouTube Music. This source means that were are limited to between and 10,000 and 20,000 training examples for the energy, valence, and vocalness classifiers and around 1,000 manual labels per genre for the multi-label genre classifiers.<sup>2</sup>

To allow robust training, even from this small amount of data, we train comparatively small, separate, fullyconnected neural nets on top of the (frozen) audio embeddings given by [6]. This is a version of transfer learning but we do not attempt to fine-tune the underlying audio embeddings for our semantic-attribute task, due to the comparatively small amount of training data we have available in our attribute space.

We then create an attribute embedding space using the continuous-valued outputs of the final logits of each of these classifiers, followed by pooled-variance normalization, as will be described in Subsection 3.2.

Details about the classifier network architectures and the training data are given next.

# 2.2.1 Genre

The genre model is a multi-label classifier (outputting 0 or more labels per video) from a vocabulary of 54 genres. It is trained on a mix of 50,000 manually labeled videos and 6,500,000 labels inferred from the DDEX feed delivered by music labels.<sup>3</sup> The genre-classification network is fully connected with 8 512-wide hidden layers. The input features are:

- Average audio embeddings [6] (across the full track)
- Average video embeddings [7] (across the full track)
- Image embedding of the video thumbnail [8]
- Word embeddings derived from a CBOW model [9] trained on 10B search queries. They are applied to the tokenized title, free-text DDEX genre, and free-text music label name of the video.
- Inferred language of the title [10, 11]
- Video type (Art Track [12], official music video, user-generated content)

While we might have been able to achieve higher music-genre accuracy by learning cross-modal coembeddings [13–15], we instead use video, image, and word embeddings trained for general video retrieval, without restriction to music-related content. This allows us to re-purpose more general embeddings, again avoiding the overhead of large-network training, just as we have with re-purposing the audio–co-listen embeddings [6].

We trained a genre-classifier network for 2 million steps using Adagrad with a learning rate of 0.05 and a mini-batch size of 64. We did not use regularization. We selected perclass thresholds by choosing the point that maximizes F1 on a separate test set.

<sup>&</sup>lt;sup>1</sup>We restricted our genre set since our evaluation was focused on playlists generated primarily for the US market and since the definition/assignment of genre is not uniform across the globe.

<sup>&</sup>lt;sup>2</sup> For genres, we also have other sources of label data, as described in Subsection 2.2.1 but that secondary source is significantly less reliable.

<sup>&</sup>lt;sup>3</sup> The DDEX feed labels must be mapped from the often idiosyncratic genre tags provided by each music label to the 54 genre label set that forms our vocabulary. That mapping is difficult to correctly determine, resulting in noisy training data. We do not use this secondary source of label data at all in evaluation.



Figure 1. Temporal Computed Energy for *Stairway to Heaven*.

#### 2.2.2 Energy

Energy is computed using the output of a regression model using full-track-average audio embeddings [6] as its only input. It was trained on around 20,000 human-labeled examples with ratings in one of 3 buckets (i.e., low, medium, high). These ratings are then converted to scores of 0,  $\frac{1}{2}$ , and 1. The network is fully connected with 2 hidden layers: the first layer is 128 units wide and the second, 64 units wide. It used Adam optimizer with a decay rate of 0.98 and was trained for 10,000 steps.

#### 2.2.3 Valence

Valence is computed using the output of a regression model, again using full-track-average audio embeddings as its only input. The network is fully connected with 3 hidden layers: the first and third layers are 256 units wide and the second, 512 units wide. It was trained on 10,500 human-labeled examples. As with energy, the training data was human bucketed ratings with 3 distinct levels, from negative (sad or angry), neutral, to positive (happy or content) and the buckets were assigned to 0,  $\frac{1}{2}$ , and 1. It used Adam optimizer with a decay rate of 0.96 and was trained for 10,000 steps.

#### 2.2.4 Vocalness

Vocalness is computed using the output of a binary classification model, using full-track-average audio embeddings as its only input. It was trained on 18,000 human-labeled examples. The raters were asked to indicate if there were significant lyrics or other vocal elements in the track. Like valence, the network is fully connected with 3 hidden layers: the first and third layers are 256 units wide and the second, 512 units wide. It used Adam optimizer with a decay rate of 0.96 and was trained for 10,000 steps.

#### 2.3 Temporal Inference

While the attributes listed above are often used to describe the entirety of a music track, there can be significant variations in some attributes over the temporal extent of a song. As an example, Figure 1 shows computed energy for the song *Stairway to Heaven*. Generating a single audio embedding representing the whole track via the mean of window samples results in a loss of information on this aspect.

For attributes like this, we shift to performing inference on 10-second segments of audio, using the time-localized audio embeddings. We do this in two steps. As a first step, we train a full-track model of the desired attribute, with the *track-level average* of the local audio embeddings as its input. With that trained model in hand, we reuse it, running a separate inference on each 10-second audio embedding, to generate time-localized attribute estimates. From this sequence of localized estimates, we compute a track-level estimate using an aggregate heuristic. For example, for the track-level energy, we take the maximum over the moving average of this temporal estimate as follows:

$$E = \max_{0 \le i < N-W} \frac{1}{W} \sum_{j=i}^{i+W-1} e_j$$
 (1)

where N is the the number of 10-second segments in a track,  $e_j$  is the raw energy estimate for the  $j^{th}$  segment, and W is the window size which also a function of N according to  $W = \max\{3, \frac{N}{6}\}$ .

In the future, we could train a time-localized regression using explicit labels on the 10-second segments or using the temporal estimates provided by the full-track model as weak labels. However, for the purpose of this paper we restrict ourselves to the method described above.

Separately, using the sequence of local estimates, we measure the attribute's consistency. The local estimate is first smoothed, to give more reliable local estimates of the attribute. For example, for energy, we use a moving average with windows as described above. From that sequence, we can create a consistency measure using:

Consistency = 
$$1 - \frac{\sum_{i=0}^{N-2} |A_{i+1} - A_i|}{\sum_{i=0}^{N-2} A_i}$$
, (2)

where  $A_i$  is the attribute value, determined by the smoothed data and centered on the  $i^{th}$  10-second segment of the track.

We improved precision from 85% to 90% using this approach instead of inference on the mean audio embedding, demonstrating the performance improvement possible from aggregating local inferences compared to performing inference on pre-aggregated embeddings. The attribute's consistency measure is also separately useful: for example, it can give playlist curators a deeper description of the acoustic-energy profile, which is needed for tasktargeted playlists like "workout" or "focus".

While this approach can be used for other (non-genre) attributes as well, for the results in this paper, we only used it with energy.

# 2.4 Accuracy of Individual Attribute Detectors

Table 1 describes the accuracy of each of our attribute models.

Energy and valence are regression models. The test set was created from human annotations on a four-point scale. The regression results were evaluated using error thresholds, set according to what was judged acceptable by the curators. Even though the training data for valence used a four-point scale, we used an evaluation threshold that is equivalent to a three-point scale. This coarser scale for valence evaluation was based on the needs of the curators.

Vocalness was trained as a binary classification and was evaluated accordingly.

Attribute	Metric	Quality
Genres: Multi-label	Human-expert	78% precision,
classifier. <sup>†</sup>	labels <sup>†</sup>	84% recall
Valence (regression,	Prediction $< 0.33$	78% accuracy
$output \in [0, 1])$	from label <sup>‡</sup>	
Vocalness (binary	Human-labeled	97% precision,
classifier)	ground truth	78% recall
Energy (regression,	Prediction $< 0.25$	90% accuracy
$output \in [0, 1])$	from label <sup>‡</sup>	

<sup>†</sup> The genre classifier was evaluated on the 16 genres used in this paper. The evaluation set was formed from the entries from expert-curated single-genre playlists and their labels were inferred accordingly.

<sup>‡</sup> The thresholds for accuracy were set after consulting with expert human curators who provided musical examples of differences in valence and energy that should be distinguishable.

 Table 1. Accuracy of Each Full-Track Attribute Model.

For genre, we created an evaluation set using entries from single-genre playlists authored by curators, with labels inferred accordingly. None of these curated tracks were used in training the genre classifier.

We are able to extract semantic attribute labels from the frozen audio embeddings, both for genre (similar to [6]) and for more qualitative measures (energy, valency, vocalness). For the qualitative measures, this labeling is based solely on the audio embedding.<sup>4</sup> Huang et al. [6] report similar findings. It is somewhat surprising that the semantic information needed to compute these labels are captured by embeddings trained for a completely different task (that is, predicting which songs are listened to together). Based on this observation, we hypothesize that an embedding space formed from these semantic attributes can be used for recommending additions to human-curated playlists. The results that we report in Section 3.2 support this conjecture and strengthen it by suggesting that our attribute-embedding space is better suited for playlist recommendations than the full audio-embedding space.

#### 3. EXPLORING CURATED PLAYLISTS

In this section, we examine the use of our learned musical attributes as a tool for discriminating between and adding to music playlists. In Subsection 3.2, we compare the discriminative power of the attribute-embedding space to that of the audio-embedding space, using the playlists that we describe in Subsection 3.1. Subsection 3.3 then describes a human evaluation of the attribute-based recommendations for playlist extension.

# 3.1 Corpus of curated playlists

YouTube Music [16] offers playlists with specific themes. Playlist themes are relevant to targeted users or markets and mostly fit into one of the following categories: ephemeral (*e.g.*, event based), seasonal, and canonical. The canonical category includes contextual (*e.g.*, bedtime music), mood (*e.g.*, feel-good favorites), and activity (*e.g.*, workout essentials) playlists. The canonical category is the most amenable to attribute analysis and automatic augmentation. For this paper, we focused on playlists in the canonical category.

We used several of curated canonical playlists to evaluate how well our attributes characterize and discriminate between the groupings that were created by human experts. Curated playlists are widely served across YouTube Music. All of these curated playlists were publicly available as of April 2020.

For our embedding-space studies, described in Subsection 3.2, we use 17 different human-curated, genre-based playlists. That combined set of playlists had 4,563 entries. To avoid evaluation-set contamination, none of the playlists or their entries were used in the attribute training (Section 2). The names of these 17 playlists are given in Figure 2 and a more complete description is given in [17].

For our playlist-extension studies, described in Subsection 3.3, we use another disjoint set of 5 different humancurated, *vibe*-based playlists. The combined set had 545 entries. General characterizations of these 5 playlists, along with their URL links, are given in [17].

#### 3.2 Analyzing playlists in attribute space

In this section, we investigate how well two different embedding spaces capture the structure of human-curated playlists. The first embedding space is the 128-dimension space generated from [6], renormalized according to the methodology described below. The second is a (renormalized) embedding space formed using the continuous logits that are trained to provide our attribute labels: that is, energy, valence, vocalness, and 16 top-level genres that are typical of music listened to in the US.

We renormalize each space using the Tikhonovregularized [18] square-root inverse of the pooled variance matrix [19]. The pooled variance matrix is estimated using a sampling of playlists and treating each playlist as a separate cluster (with an independent cluster mean) but with a single shared (pooled) variance matrix. The embedding space is then rotated and scaled, according to the squareroot inverse of this pooled variance. We use Tikhonov regularization in this inversion to avoid possible problems with nearly singular variance matrices.<sup>5</sup>

After re-normalization, each playlist is (on average) a Gaussian distribution with an identity-variance matrix, allowing us to directly compare between-playlist distances across the two embedding spaces. We use this distance equivalence throughout this section, to determine how well human-curated playlists are separated in these two embedding spaces. Our hypothesis is that, whichever embedding space gives better separation between authored playlists will also give better suggestions for creating or extending playlists. We will more directly examine how well our suggestions do for playlist generation in the next section. Before moving to that analysis, we compare the mathematical performance of the two spaces in this section.

<sup>&</sup>lt;sup>4</sup> For genre, we provide our networks with information derived from the video content and text in the title and description (see Subsection 2.2.1), as well as the co-listen–based audio embeddings.

<sup>&</sup>lt;sup>5</sup> We did not observe any near singularities in either the attribute- or the audio-embedding spaces but continued to use it, to avoid issues in the future, when we plan to use a larger group of attributes.



(a) using normalized attribute-embedding space

(b) using normalized audio-embedding space

Figure 2. Average over each playlist k of  $d_{i,k,j}$  (defined in Eqn (3)). See right column for the mapping from the playlist numbers to their titles and [17] for their descriptions and URL links.

We compared the separation of the 17 different humancurated, genre-based playlists from Subsection 3.1 by examining the distances between each playlist entry and all of the playlist centroids:

$$d_{i,k,j} = ||e_{i,k} - m_j||^2$$
(3)

where  $e_{i,k}$  is the embedding-space coordinates for the  $i^{th}$ entry in the  $k^{th}$  playlist and  $m_j$  is the mean of embeddingspace coordinates across all  $N_j$  entries in the  $j^{th}$  playlist:  $m_j = \frac{1}{N_j} \sum_{i=0}^{N_j-1} e_{i,j}$ . Figure 2 shows the average of these distances for each playlist: that is,  $\frac{1}{N_k} \sum_{i=0}^{N_j-1} d_{i,k,j}$ . The larger the distance the less "alike" the two playlists appear in that embedding space. Based on Figure 2, the (normalized) attribute embedding space does a better than the (normalized) audio embedding space at separating the playlists, while keeping each individual playlist compact.

We can look at this separation/compactness of playlists in each embedding space, with one summary statistic per playlist entry. We use  $\Delta_{i,k} = \min_{j \neq k} d_{i,k,j} - d_{i,k,k}$ : the smallest difference between each entry's distance to the closest, "other" mean and its distance to its own mean. Figure 3 shows the histograms of this relative distance measure for the (normalized) attribute-embedding space and the (normalized) audio-embedding space, as well as the histogram of the paired difference between them. For Figure 3-a and Figure 3-b, highly positive values are best and negative values indicate an entry that is closer to a different playlist's mean than to its own. For Figure 3-c, positive values correspond to the attribute-embedding space giving better separation than the audio-embedding space. Using a single-tail, paired Student t-test [20] on this data indicates that the attribute embedding space is significantly better than the audio embedding space with a probability well over 99% (t = 27.24, p = 3e-151). In order to be certain that this high level of significance does not derive from unequal variances across the two embedding spaces, we also ran a single-tail Welch's unequal-variance t-test [21]: this still showed well above 99% certainty (t = 16.24, p = 5e-58).



14

15

16

South Side Nights

Intro to Haydn

Reggae-Pop Crossovers

**Figure 3**. Histograms of  $\Delta_{i,k}$  using the two different embedding spaces and of their difference.

#### 3.3 Extending playlists using attributes

In this section, we explore the application of the learnedattribute space to generating algorithmic candidates that could be used to refresh or extend human curated playlists in the corpus described in Subsection 3.1. One advantage of using the learned-attribute space is the ability for humans to understand, debug, and tweak the automatic method. We run an experiment where we show professional music curators a set of candidates for a playlist and ask them to assign a rating of whether those candidates sufficiently align with its vibe.

#### 3.3.1 Playlist selection

We selected 5 playlists from the corpus for this experiment with the premise that they had consistent vibe with focus on context/mood/activity rather than, for example, most popular or new releases. We hypothesize that a semantically meaningful attribute space would be better at generating recommendations closer to the vibe of such playlists compared to traditional co-listen-based approaches.

#### 3.3.2 Candidate generation

For each playlist, we aggregated attribute scores across its tracks to create a *recipe* consisting of the following:

• Top-N genres contributing to 80% of the cumulative frequency distribution where a genre is assigned to a playlist if its score was above the threshold determined from evaluation in Subsection 2.4.

Playlist	Rating			Total
	Good	Borderline	Bad	Total
Classical for Sleeping	36%	38%	26%	214
Classic Sunshine Soul	39%	35%	26%	101
Tranquil Spa Day	37%	63%	0%	27
Feeling Good in the 80's	22%	20%	58%	143
90's Rock Relaxation	11%	24%	65%	85

**Table 2.** Music-curator ratings on recommendations forplaylist extension. See [17] for the description and URLlink of each playlist.

- Mean and standard-deviation of the real-valued attributes energy, valence, vocalness. We also included tempo in beats-per-minute computed using APM [22] as an additional attribute.
- Other metadata attributes including top-10 artists, earliest and latest release year of tracks on the playlist. These were added as constraints to weed out recommendations too far away from the playlist premise.

This *recipe* is then used to generate a list of tracks classified into the top genres, with real-valued attribute scores at most one standard-deviation away from mean, release year within the earliest and latest release year and performing artist(s) among the top artists. We sort this list by popularity in the last 365 days and prune to generate a final list of recommendations.

#### 3.3.3 Curator assignment

Music curators were shown these recommendations and asked to assign a rating from below options

- Good "track is not only appropriate for the playlist premise, but also a high-quality recommendation"
- Borderline "while track's attributes align with the premise, I would not be excited to program it"
- Bad "I would never program this track to this playlist, because it does not fit the premise". Raters were also asked to note a reason in this case.

#### 3.3.4 Results

The results of the experiment are tabulated in Table 2. We find that, for playlists defined almost solely by mood and emotional affect, the curators found a majority of the tracks good enough to program onto the playlists and some "bad" tracks. For *Tranquil Spa Day* especially, there were no "bad" tracks in the 27 that were rated. This shows that the recipe based on semantic attributes and metadata constraints was a decent heuristic for playlist extension.

For the decade playlists (last 2 rows) the performance was very poor. To have a better understanding, we analyzed the rater notes and found that 77 out of 83 and 45 out of 56 "bad" tracks for *Feeling Good in the 80s* and *90's Rock Relaxation*, respectively, were due to the curators not feeling that the tracks belonged to the correct decade. On

examination, we found that the metadata was indeed incorrect on those tracks. Discounting these tracks with incorrect metadata, our approach again seems to perform decently on these playlists.

For a qualitative study like this, the strongest support is the overall evaluation by the music curators on whether or not the suggestions are useful to have. Even with around one-in-four playlist suggestions being discarded as incorrect, the music curators found that having these automatically generated suggestions available sped up their work on refreshing and extending the vibe-oriented playlists.

# 4. CONCLUSIONS

We described a system and method to automatically label musical tracks with semantically meaningful attributes, including musical genre, autonomic arousal, valence, and vocalness. These attributes are inferred using models operating on audio embeddings generated by deep neural networks trained on co-listen data, using triplet loss. The attribute models themselves are trained using smaller amounts of labeled data. We show that precision improvements can be obtained by running attribute inference on temporal segments and fusing those scores into a wholetrack score compared to running inference on an averaged embedding. This approach also yields temporal consistency attributes that are useful in and of themselves.

We then define a lower-dimensional embedding space established by these semantic musical attributes. We compare these embeddings with the original audio co-listentrained embeddings in the context of professionally curated playlists. We find that this space better separates a sample of thematic playlists: it matches the semantic similarity implicit in these professionally curated playlists better than the raw audio embedding space.

Unlike previous studies of playlist extension [23, 24], we used these semantic attributes to generate humanreadable and -editable recipes for professionally curated playlists. We used those recipes to automatically extend the playlists and measured the quality of those automatic content refreshes via human evaluation.

#### 5. REFERENCES

- J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representations for music annotation and retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2012.
- [2] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in Advances in Neural Information Processing Systems. Curran Associates, Inc., 2013.
- [3] P. Hamel, M. Davies, K. Yoshii, and M. Goto, "Transfer learning in MIR: Sharing learned latent representations for music audio classification and similarity," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2013.

- [4] A. van den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Proceedings of* the International Society for Music Information Retrieval Conference, 2014.
- [5] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2017.
- [6] Q. Huang, A. Jansen, L. Zhang, D. Ellis, R. Saurous, and J. Anderson, "Large-scale weakly-supervised content embeddings for music recommendation and tagging," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2020.
- [7] S.-J. Hwang, J. Lee, B. Varadarajan, A. Gordon, Z. Xu, and A. Natsev, "Large-scale training framework for video annotation," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.
- [8] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of International Conference on Learning Representations Workshop*, 2013.
- [10] R. Datta, "PHIL: The probabilistic hierarchical inferential learner," in *10th Annual Bay Area Discrete Mathematics Day*, 2005. http://math.berkeley.edu/~datta/philtalk.pdf.
- [11] T. Jauhiainen, M. Lui, M. Zampieri, T. Baldwin, and K. Lindén, "Automatic language identification in texts: A survey," *Journal of Artificial Intelligence Research*, vol. 65, 04 2018.
- [12] Y. Help, "What is an Art Track?" https://support.google.com/youtube/answer/6007071, 2020.
- [13] D. Surís, A. Duarte, A. Salvador, J. Torres, and X. G. i Nieto, "Cross-modal embeddings for video and audio retrieval," in *Proceedings of the European Conference* on Computer Vision Workshops, 2018.
- [14] B. Li and A. Kumar, "Query by video: Cross-modal music retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2019.
- [15] B. Elizalde, S. Zarar, and B. Raj, "Cross modal audio search and retrieval with joint embeddings based on text and audio," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

- [16] Y. Music, "https://music.youtube.com," 2020.
- [17] U. Gargi, A. Patwari, N. Kong, and J. Wang, "Playlists used for semantically meaningful co-listen attributes from embeddings playlist exploration and expansion," for in https://github.com/YTMCC/ismir2020/blob/master/ playlists.md, 2020.
- [18] A. Tikhonov and V. Arsenin, Solution of Ill-Posed Problems. New York: Halsted Press, 1977.
- [19] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley Interscience, 2001.
- [20] R. Walpole, R. Myers, S. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*. Boston: Prentice Hall, 2012.
- [21] G. Ruxton, "The unequal variance t-test is an underused alternative to Student's t-test and the Mann-Whitney U test," *Behavioral Ecology*, vol. 17, no. 4, 2006.
- [22] D. Eck, "Beat tracking using an autocorrelation phase matrix," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.
- [23] C.-W. Chen, P. Lamere, M. Schedl, and H. Zamani, "Recsys challenge 2018: Automatic music playlist continuation," in *Proceedings of the ACM Conference* on Recommender Systems, 2018.
- [24] P. Papreja, H. Venkateswara, and S. Panchanathan, "Representation, exploration and recommendation of playlists," in *Proceedings of the European Conference* on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Workshops, 2019.