ZERO-SHOT CRATE DIGGING → DJ TOOL RETRIEVAL USING SPEECH ACTIVITY, MUSIC STRUCTURE AND CLAP EMBEDDINGS

Iroro Orife
iroro@alumni.cmu.edu

ABSTRACT

In genres like Hip-Hop, RnB, Reggae, Dancehall and just about every Electronic/Dance/Club style, DJ tools are a special set of audio files curated to heighten the DJ's musical performance and creative mixing choices. In this work we demonstrate an approach to discovering DJ tools in personal music collections. Leveraging open-source libraries for speech/music activity, music boundary analysis and a Contrastive Language-Audio Pretraining (CLAP) model for zero-shot audio classification, we demonstrate a novel system designed to retrieve (or rediscover) compelling DJ tools for use live or in the studio.

1. INTRODUCTION

When DJs are mixing music, or remixing specific songs, creating special edits, re-drums, mashups or just long-playing mixtapes, DJ tools provide a host of creative possibilities. Tools vary by genre and era, but are generally short, simplified musical phrases retrieved from existing music with the intention to reuse in a DJ performance. These musical phrases range from sound effects to acapella loops to purely instrumental passages, solo percussion or drum break to an entire verse or bridge of song. For example a DJ might trigger an acapella loop or long sound effect while mixing a transition from SongA → SongB. DJ Tools are commonly sold in online shops along with royalty-free sound libraries, sample packs of loops and beats. Most tools include key signature, beat and tempo metadata where necessary to ensure sync to the DJ project master tempo.

1.1 Crate digging & a short history of DJ tool

Before the advent of online shops trading sonic tools, DJs and producers were known to spend time in record shops crate-digging, or hunting for rare, vintage, or otherwise obscure vinyl with interesting breaks, melodic hooks, drops, intros/outros, or B-side acapellas. Practise time was devoted to studying the structure of music, identifying suitable mix points, curating tools and experimenting with different creative interpolations between two mixable songs.

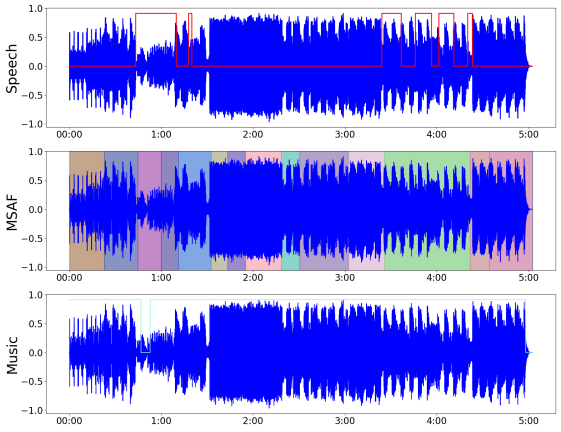

Figure 1. A 5 minute Ragga Jungle song overlaid with detected speech and music activity, as well as music-structural boundaries

Then while playing live, tools are triggered or looped from Sampler modules or "Remix Decks" connected to the DJ mixing board.

1.2 Musical structure, speech activity and zero-shot classification

DJ tools naturally occur at moments in a song where there is a transition to a simpler, less-dense mix. Ergo, we leverage the music structural analysis framework (MSAF) boundary detection algorithms to supply the approximate time-offsets of structural progressions [1]. Next, we employ a speech and music activity detector (SMAD) [2] to further refine the selection of suitable passages.

For DJ-tool classification we engage the zero-shot capabilities of a Contrastive Language-Audio Pretraining (CLAP) model [3]. Given an audio segment X^a and a list of text descriptions of different DJ tool classes $\{X_1^t, ..., X_M^t\}$, we use CLAP's pretrained {audio, text} encoders and their projection layers to compute CLAP embeddings E^a and E_i^t. The classification logits D_i can be computed as magnitude of the cosine similarity between the audio segment embedding and *each* text embedding.

$$\begin{aligned} X^a &\rightarrow AudioEncoder \rightarrow E^a \\ \{X_1^t, ..., X_M^t\} &\rightarrow TextEncoder \rightarrow \{E_1^t, ..., E_M^t\} \end{aligned} \quad (1)$$

$$D_i = Similarity(E^a, E_i^t) \quad (2)$$

Late-Breaking / Demo Session Extended Abstract, ISMIR 2024 Conference

2. THE SYSTEM

The system, its libraries, data and model dependencies, are all fully open-source software (OSS) written in Python. [1] Firstly, we run MSAF [2] and SMAD [3] to generate raw activities and structural boundaries, saved as CSV files. Next, these are post-processed to yield lists of time ranges and boundary times. In principle, we can just use the boundary times to segment the song, however using speech activity onset times delivers more precise slices. Figure 1 shows the relationship between the boundaries and the speech activity for one song. For zero-shot classification, we utilize the LAION version of CLAP [4] hosted on Huggingface [4]. Finally, requiring some creativity and trial-and-error, we manually create a list of descriptive text strings for each class. A few abbreviated DJ tool descriptions are listed in Table 1.

2.1 Algorithm

Below we outline how we go about processing all files in a music library. W^s, W^m are "windows" or time ranges specified with a `start_time, end_time, label`

Algorithm 1: Zero-shot Crate Digging

Create text prompts for M classes $\{X_1^t, ..., X_M^t\}$
$\{E_1^t, ..., E_M^t\} \leftarrow TextEncoder(\{X_1^t, ..., X_M^t\})$
for *song* s_i *in the library* S_N **do**
  $W^s, W^m \leftarrow SMAD(s_i)$
  $B \leftarrow MSAF(s_i)$
  Adjust boundaries B_i to nearest W^s
  Use B to cut s_i to audio files $\{X_1^a, ..., X_N^a\}$
  for $j :=$*1 to* N **do**
    $E_j^a \leftarrow AudioEncoder(X_j^a)$
    $\vec{D} \leftarrow Similarity(E_j^a, \{E_1^t, ..., E_M^t\})$
    $\vec{z} \leftarrow softmax(\vec{D})$
    Store predicted class, the $\arg\max$ of $\vec{z}$

2.2 Evaluation and discussion of limitations

To evaluate the system, we generated classifications from the author's DJ library. Overall, the system performs well for vocal and percussive tool classes with "prediction probabilities" > 0.75. However, shorter burstier sound effects, drops and genre-specific tools failed to be identified consistently. As expected, the system is sensitive to *precise language* and we observe that adding broad or generic terms or entire genres (e.g. "hip-hop, funk, drum and bass") to a single class description led to markedly worse results across all classes. For best results, the classes should not overlap in their description, but can contain mixed elements. For example our best performing vocal description was "acapella, expressively sung human vocal with background instrumental music tracks".

[1] https://github.com/ruohoruotsi/djtool-crate-digging
[2] https://github.com/urinieto/msaf
[3] https://github.com/biboamy/TVSM-dataset/tree/master
[4] https://huggingface.co/laion/clap-htsat-unfused

DJ Tool class	Example text description
Acapella loops	"expressively sung vocal tracks"
Sound effects	"siren, riser sound effects, whoosh"
Drums breaks	"drum beat, drum solo, breakbeat"
Melodic hooks	"strings, solo guitar, piano melodies"
DJ Drops	"a high energy, massive EDM drop"
Battle tracks	"vinyl scratch loop, turnatablism"

Table 1. In practise, text descriptions are more tortuous

Sliced segment	23s	18s	13s	8s	3s
01_vox.wav	1.	1.	0.99	0.25	0.01
02_drums.wav	1.	1.	1.	1.	1.
03_vocalhook.wav	1.	1.	1.	0.99	0.96

Table 2. Segment predictions, varying segment duration

DJ tool durations are typically at the motif or phrase level, i.e. longer than a single utterance, note or beat but shorter than an entire verse or chorus. The challenge was to determine optimal segmentation for CLAP classification, while slicing the song appropriately for use by DJs. So we ran an experiment, using three 23s segments from ground-truth-ed songs [5,6], tracking how the predicted class probabilities varied as we progressively reduced the segment duration to 3s. We observe that for tools with just one class {*02_drums.wav*, *03_vocalhook.wav*} featuring breakbeats and purely sung vocals respectively, that predictions are stable even at short durations. Whereas for *01_vox.wav* which includes vocals, stuttery vocal effects and eventually background music, its predictions are more sensitive to local features and variances.

3. RELATED WORK

There is not much existing work within the community relevant to DJs. The few studies that we found focused on Pop or EDM genres, which have much less of a crate-digging history. These works focused on the challenges of ideal sequencing of songs and inter-song transitions, with an eye toward the dream of a fully automatic DJ [7–9].

In a recent work, DJ StructFreak (2023), the authors tackle the task of carefully choosing suitable "mix points" *within* a song, to generate smooth, pleasing transitions [10]. Similar to our work, they employ music structural analysis and use embeddings from a pretrained model [11].

4. FUTURE WORK

There is much work to be done to develop the system into a tool for non-programmer disc jockeys. Future algorithmic work includes improving the fidelity of the boundary detection algorithms and evaluating recent structural segmentation approaches [11]. Finally, if there is enough interest to turn DJ tool retrieval into a proper MIR task, then we will need labeled datasets.

5. REFERENCES

[1] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research." in *ISMIR*, 2016, pp. 547–553.

[2] Y.-N. Hung, C.-W. Wu, I. Orife, A. Hipple, W. Wolcott, and A. Lerch, "A large tv dataset for speech and music activity detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, p. 21, 2022.

[3] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap: Learning audio concepts from natural language supervision," 2022. [Online]. Available: https://arxiv.org/abs/2206.04769

[4] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[5] DJ. Ash and DJ. Vern, "Squeeze," in *Toasted: Massive Ragga Jungle*. New York City, USA: Mutant Sound System, 1998, ch. 12.

[6] 91Vocals, "Vocal packs, premium vocal hooks," https://91vocals.com/en-us/collections/vocal-hooks, accessed: 2024-10-04.

[7] A. Kim, S. Park, J. Park, J.-W. Ha, T. Kwon, and J. Nam, "Automatic dj mix generation using highlight detection," *Proc. ISMIR, late-breaking demo paper*, 2017.

[8] Y.-S. Huang, S.-Y. Chou, and Y.-H. Yang, "Djnet: A dream for making an automatic dj."

[9] R. M. Bittner, M. Gu, G. Hernandez, E. J. Humphrey, T. Jehan, and N. Montecchio, "Automatic playlist sequencing and transitions."

[10] T. Kim and J. Nam, "Dj structfreak: Automatic dj system built with music structure embeddings," in *Ismir 2023 Hybrid Conference*, 2023.

[11] ——, "All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio," in *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2023, pp. 1–5.