Natural Language Processing Enabled Concatenative Audio Synthesis

Student ref. 33655491

Topic

Concatenative audio synthesis has a long history of introducting ML methods to creative sound practices. Corpus-Based Concatenative Synthesis toolboxes such as IRCAM's *CataRT)* [1] as well as *FluCoMa*[2] developed at University of Huddersfield both provide advanced tools for audio segmentation, analysis and re-synthesis. Currently, similar ways of thinking are applied to Deep Learning and Neural Audio Generation, sharing some, but not all of the methodology.

Corpus Analysis

Both of the above are based on decomposition of an audio signal into short grains, which then undergo feature extraction, organising a granularized sound corpus based on either so called *crude* features (ie. loudness, estimated pitch) or more complex ones (such as novelty, chroma, MFCCs or spectral shape). Since each grain can be represented as a vector of N-dimensions), machine learning techniques are commonly applied to reduce the dimensionality of the feature space to 2D, the end goal being an intuitive and interactive exploration of sound organised on a plane. This type of approach is commonly termed 2D Corpus Exploration.

One of its characteristics is its almost complete indifference towards the sequential character of the original input on any scale, as even features relating directly to the pitch domain are usually averaged per-grain.

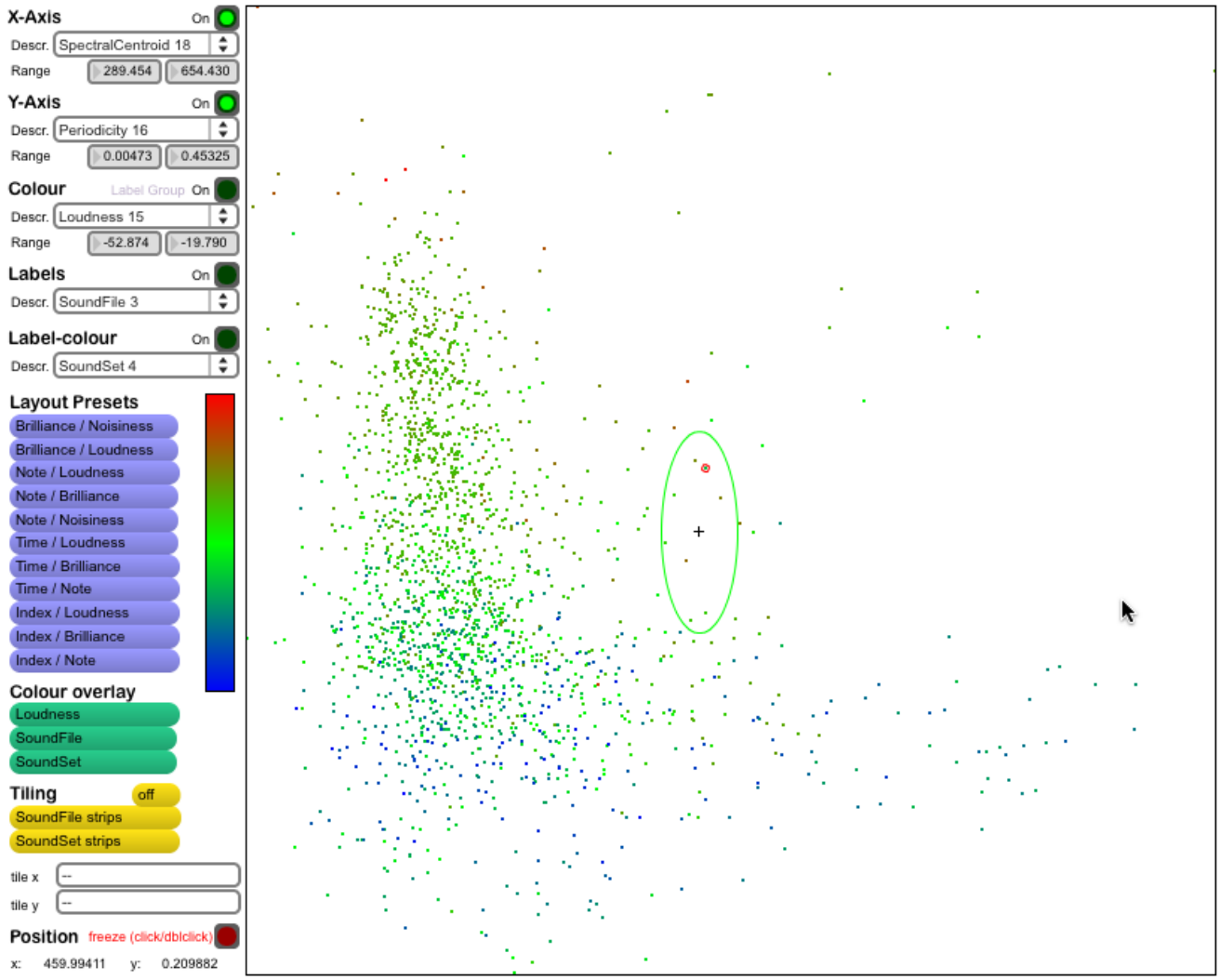

Fig. 1 - *2D display of catart.lcd5*

Deep Neural Networks

Auto-Encoders

More recently, neural networks based on auto-encoder architecture have proved themselves capable of audio generation with almost-approachable performance requirements with their implementations such as *Musika!* coming out of University Linz and *RAVE* from ACIDS team at IRCAM (again), both possible to train on consumer grade GPUs.

Both of these deal with compactly encoding (spectral or other) representations of audio frames (which are indeed segmented, similar as in Corpus Analysis), and then modelling that representation with the purpose of reversing the process, enabling generation of audio of arbitrary length.

Musika!, which claimingly works faster than real-time[3] does not happen to have an actual real-time implementation - the only enabled way of generating audio with a trained model

is by using a Python script. In case of *RAVE* (real-time is what the "R" character in its name actually stands for), models can be wrapped using Torchscript and be interacted with, be that via a prior latent model trained on top of the latent representation, or via direct interaction with the latent vector.
Interestingly, even such advanced software as this still is rather agnostic towards longer-term structure. Generated grains might be clustered together in an order that forms a longer, recognizable sound (ie. drum hit, or a spoken syllabe), but they are forgotten about as soon as they leave the buffer.
At the same time, they still do require substantial amounts of training data and prove way more computationally expensive than the *classic*, sample-based concatenative approaches, considering they synthesize audio samples from scratch.

Transformer Natural Language Processing

Another angle had also been considered as a part of OpenAI's *MuseNet*[4] project, which applies transformer-based sequence prediction (widely used for natural language processing) to generate MIDI patterns. In this case, the only interaction with the network happens through an online interface, which allows a limited range of parameters: style (based on composers included in the database), intro sample to base the generation on, instrumentation and a number of tokens to be generated. It is able to extend existing sequences in a comprehensive manner most of the times, with an exception for unexpected pairings of instruments and styles (ie. Chopin-style arrangement for drums) and has a broadly understood sense of placing events in time, compared to the previous examples. At the same time - *MuseNet* does not have any capabilities of sound synthesis nor real-time usage.

Research question

The basis for my inquiry is the possibility of combining some aspects of the forementioned approaches (grain-based analysis, concatenative synthesis, Natural Language Processing), creating a versatile tool for audio processing and generation, with focus on goals such as:

- processing and re-synthesis of arbitrary audio signal,
- coherence of output on both micro and macro scale (be it grains or whole sections of input),
- intuitive real-time interaction,
- low performance requirements.

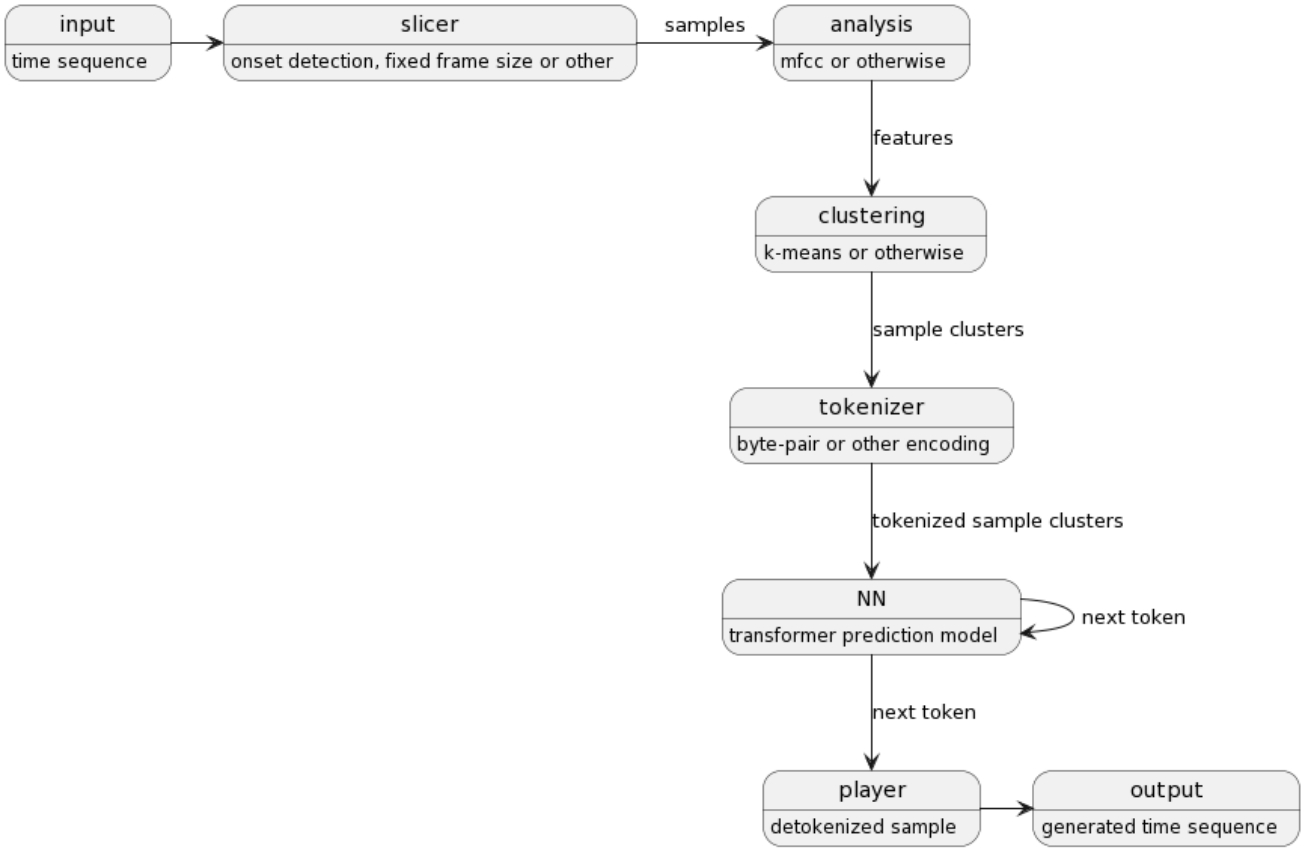

Fig. 2 - *tool prototype data flow diagram*

Project timeline

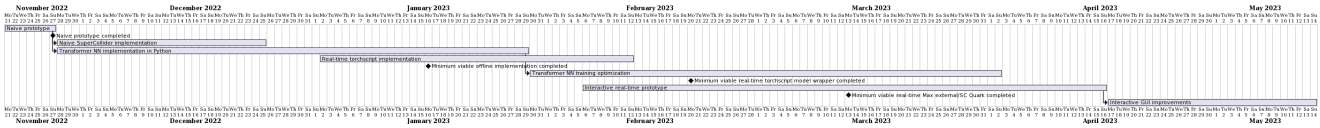

Fig 3. - *project timeline*

---

1. Schwarz, D., Corpus-Based Concatenative Synthesis : assembling sounds by content-based selection of units from large sound databases. IEEE Signal Processing. Mars 2007, vol. 24, n° 2, p. 92-104↩
2. Tremblay, P.A., Roma, G., & Green, O. (2022) Enabling Programmatic Data Mining as Musicking: The Fluid Corpus Manipulation Toolkit. Computer Music Journal 2022; 45 (2): 9–23. doi: /10.1162/comj_a_00600↩
3. Pasini, M., Schlüter, J., Musika! Fast Infinite Waveform Music Generation, https://arxiv.org/abs/2208.08706.↩
4. McLeavey Payne, C., (2019) MuseNet, https://openai.com/blog/musenet/↩