Musical Voice Synthesis at the Midpoint: Where Text Meets Sound

Dr. Olivier Pasquet
Goldsmiths, University of London
o.pasquet@gold.ac.uk

ABSTRACT

This research explores the application of language models and machine-learning techniques to generate musical voices with personality. By utilizing autoregressive transformers, specifically the Bark model, tokens are generated from input text to produce a unique invented and undefinable language.
Composition is being made between text and sound using various control techniques, including tokens repetition and windowing, Lempel-Ziv-Welch compression, and token clustering from acoustic feature extraction, to regulate the output voice's granularity, intelligibility, and meaning. A recursive generation system is also introduced, allowing for the creation of a large series of interrelated voices.
The research is used in various artistic applications, including music remixing and theater production. It explores other forms of expressive voices and storytelling seamlessly lying right in the middle between text and sound.

1. INTRODUCTION

Many have previously played with Text-To-Speech (TTS) synthesis engines by typing nonsensical text as input. We even built loops and random texts, which we then used to create thousands of audio files for post-processing in previous pieces using synthetic voice. The results were very satisfying, but they tended to become cliché unless we incorporated additional musical techniques afterward. We were never seamlessly working halfway between text and sound; it was naturally always a back-and-forth process between those two layers.

2. VOICE GENERATION WITH PERSONALITY

2.1 Artificial "musical" voices

Georges Aperghis's work obviously influenced our research. His music-theatre pieces defy categorization. But it belongs to a kind of vocal composition relying heavily on a virtuosic manipulation of "phonemes", marked by rapid tempos, repetitive patterns, and accumulative techniques, all of which generate intense rhythmic energy [1] [1]. The creation of an "imaginary language" gives rise to a soundscape that is both ambiguous and playfully humorous, evoking the illusion of communication while remaining music. This blurs the line between linguistic expression and musical composition.
This blurred line can usually be explored and used with symbolic text and signal processing separately. However, there have been exceptions merging those steps such as Sprechgesang or Sprechstimme's expressionist musical vocal techniques. They lie in the middle but also follow the chain of being first defined symbolically then interpreted by vocalists.
Hip-hop and R&B have continually pushed the boundaries of vocal expression through various techniques that blur the line between literalism and abstraction. Notable vocal techniques are added to audio techniques such as Autotune [2]. Since the late 1990s, Auto-tune has evolved from being merely a vocal correction tool into a cultural phenomenon. This effect is based on re-synthesis and can be extended as an artificial voice controlled by voice. This instrument is able to transform both voice itself and the meaning it conveys.

2.2 Search for a synthesis with personality

Voice synthesis controlled by a Large Model Language (LLM) allows for generating a wide range of text and vocal techniques that are different when asking a performer. Depending on the models and used techniques, synthesis can bring a wide range of variability, subversion and inspirations. Such synthesis allows emotional detachment, gender, and neutrality that we can hybridize at will.
Moreover, it allows composing at the exact place of the blurred line between literalism and abstraction; and between symbolic text and signal.
However, most synthesis engines' quality has become too good to be mere instruments. Their lack of glitches and inconsistency does not enhance the creativity of the tool. Moreover, they significantly lose character, and the voice is far less creative than that of a real actor, for instance. Finally, it is challenging to compose using voice synthesis without an architecture that we can fully control, avoiding babbling effects closely tied to early-2020s aesthetics.

3. WORKFLOW

We propose a workflow offering control at all levels between the initial text generation and the final voice synthesis. The first process of the chain consists on composing text using LLM in Ollama [3]. We then employ a voice synthesis engine named Bark to generate audio tokens, which we convert to directly utilize the FairSeq

[1] "Phonemes" does not refer to proper linguistics but rather portions of words or voice techniques.

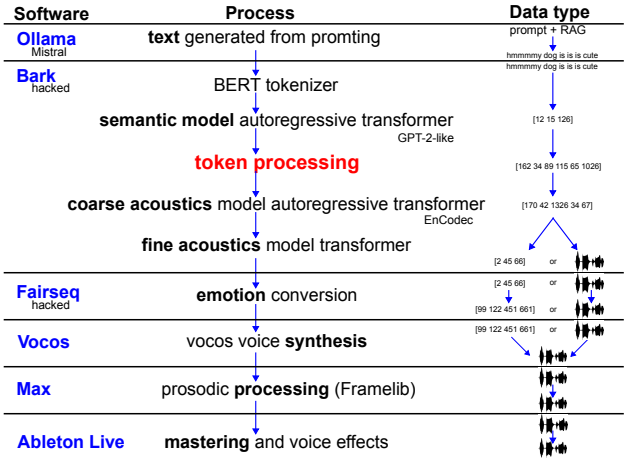

Figure 1. Proposed workflow with variables that can be utilized for musical purposes. This paper will focus specifically on the Bark part, particularly the red section, which has been demonstrated to be the most effective and expressive.

library for emotion conversion [2]. The final synthesis is done using Vocos neural vocoder and sent to Max for live prosodic post-processing and Ableton Live for mastering and voice doubling [3].
We will here focus on one part of the voice synthesis although all components are interdependent and sometimes necessarily influenced by one another. We then only concentrate on the red section shown in fig. 1 that processes "semantic token" that lie exactly where we want: in the middle between text and audio generation.

4. HACKED BARK

4.1 Adapting Bark to composition

We decided to start with an adapted version of Bark, a text-prompted Generative Pre-trained Transformer-style (GPT-style) model that takes creative liberties in its generation. Suno's program, from 2023, does not offer the best sound quality but this has no impact on the overall quality of our system, as we employ numerous other processes and syntheses afterward [4]. Moreover, the Max component at the end of the workflow significantly alters voice qualities for aesthetic purposes [5].
Bark is made of a series of autoregressive transformers using a semantic model, a coarse acoustics model, and a fine acoustics model:

- The *fine acoustics model* takes as input predicted tokens generated from the coarse model and iteratively predicts tokens ready for the audio synthesis. The use of EnCodec neural codec permits coding and hooking Bark to other libraries [6] [4].
- The *coarse acoustic model* is a GPT-2-style causal transformer converting semantic tokens into coarse acoustic ones.
- The *semantic model* is also a GPT-2-like causal autoregressive transformer model with a language modeling head on top. It takes in tokenized text (from a BERT tokenizer) as input and then predicts the semantic tokens that encode the audio to be generated. This part is the most important for speaker's identity. We can here add prompts that will most define speakers' personality traits with their intonation and prosodic patterns.

Bark's architecture is powerful for creativity thanks to its GPT architecture extending beyond only voice. Its models encompasses a wide variety of nonverbal communications like laughing, sighing, crying, and other surprises that can be called by prompting depending on the model used. However, this strength also brings the weakness of the outputs quickly getting unpredictable if not properly inferred. The maximum length of audio in Bark is 756 semantic tokens, equivalent to approximately 15 seconds. This is due to the model's context window size being capped at 1024, similar to how text language models have limited context sizes today (e.g. 4096). It's worth noting that if Bark utilized relative positioning instead of absolute positioning, it might have been possible to achieve larger context sizes, such as 2048 or 4096. However, currently, we have not found techniques for achieving this with absolute positioning.
Variable windowing of tokens allows us to control the granularity of the sound and the intelligibility of the voice. It aligns with the infinitesimal rhythmical aesthetic we envisioned in our music. However, using a set of random tokens at this stage rapidly disrupts the model's predictive capabilities, typically resulting in a degenerate output: a converged, monotonous pitch with filtering and noise (fig. 2). The choice of variable window size and the randomness of token sequences defines how much prediction there will be.

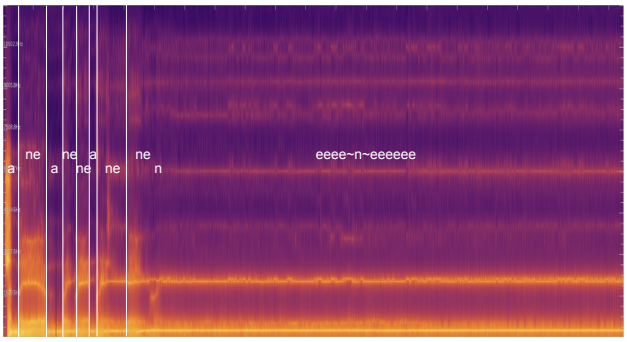

Figure 2. LPC spectrogram showing the predictive limits of the model. Using low temperature makes it diverge when not using unconventional sequences of tokens or when asking for a longer duration than what the system was designed for. We see here that it starts with the intended text then gets into a pitched loop. This inherent characteristic can be transformed into artistic control.

4.2 Random variables for musical control

Each of those three layers has the following standard controls seen in such models. These include temperature, Top-p, and Top-k controls:

- *Temperature* setting governs the degree of randomness in word selection during text generation. Lower

[2] Fairseq lib: https://github.com/facebookresearch/fairseq
[3] Vocos lib: https://github.com/gemelo-ai/vocos
[4] Unhacked Bark: https://github.com/suno-ai/bark
[5] FrameLib: https://github.com/AlexHarker/FrameLib
[6] EnCodec codec: https://github.com/facebookresearch/encodec

temperatures yield more predictable and consistent outputs, whereas higher temperatures introduce greater freedom and creativity, albeit at the cost of consistency.

- *Top p* setting determines the number of probable words considered by the model. Higher values enable the model to examine a broader range of possibilities, including less likely words, resulting in more diverse generated text.
- Adjusting the *Top k* setting influences response repetitiveness and complexity, notably in vocabulary and phrasing.

We have implemented a method to control these values using break-point functions (BPF), which enable us to regulate amounts of randomness within each sequence of given tokens. This helps play with the varying intensity of expression throughout a sentence. We added several random and quantizing engines akin to those found in Ableton Live's *Beat Repeat* feature [5]. Randomly repeating tokens this way sounds very much like granular synthesis. But employing such transformers with temperatures as described earlier yields a more dynamic and human-like output. It offers greater control and expressiveness compared to only directly concatenating grains. The articulation between "phonemes" aims for naturalness and may sometimes evoke the additive interpolation taste found in *Diphone* (1999) [6, 7].

Randomness plays a valuable role in facilitating serendipitous composition experiments, although it falls short of enabling nuanced musical sequence composition. We require greater control over token sequence generation.

5. TOKEN ENGINEERING

5.1 Markovian methodology

We subsequently generate more sentences than needed, thereby producing a large set of tokens that can then be organized simply using probabilities. The initial generated text must indeed have some consistency to retrieve it after the synthesis. We therefore use Ollama's models to produce a set of paraphrased sentences that share a sufficient number of common words, thereby facilitating our analysis [7]. This instantly allows us to reach more meaningful textual material in the output, and we are approaching the initial idea of having an invented language. We also used a various set of Markovian techniques. Our system is a mixture of probabilities and neural networks which is historically interesting.

We subsequently employed the multi-scalar Lempel-Ziv-Welch (LZ) compression, acknowledging that windowing was playing a critical role. We can decode sequences from its dictionary and sequence using a collection of probability processes [8, 9]. Using LZ with weighted probability has been demonstrated to be the most effective method for regulating the level of meaning and abstraction in our voices. The multi-scalar nature of the algorithm allows us to choose specific sequence lengths from the LZ dictionary and play with our model's context window, as described earlier.

[7] https://ollama.com - We mostly prompt Mistral models and widely use Retrieval-Augmented Generation (RAG) technique.

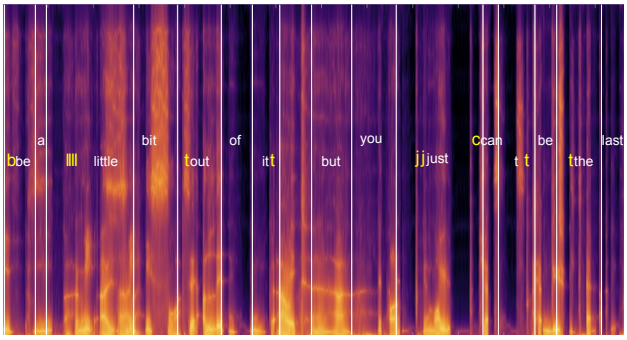

Figure 3. Example use: generation of semi-words and semi-sentences in English using an LZ sequence (white) overlapped with accidental repetitions of particular sets of semantic token (yellow).

5.2 Acoustic properties methodology

We now aim to achieve token segmentation enabling us to extract specific sound or patterns from within the model's output. The inherent unpredictability of autoregressive transformers, particularly when using models not trained by ourselves, necessitates a thorough analysis of the model's behavior: We query the model before using it. To minimize uncertainty, we employ a deterministic strategy by using low-temperature settings and a unique seed number.

We initiate our analysis with a comprehensive dataset of paraphrased sentences, which we then subject to acoustic feature extraction using the *Flucoma* Python library and simple house-made k-means clustering [8] [10]. The multilingual feature of Bark models allows for even greater timbral variety. Specifically, we analyze pitch sequences and Mel-Frequency Cepstral Coefficients (MFCCs) derived from the output, enabling us to perform pitch-based or timbre-based clustering of tokens [9]. This approach proves highly effective for token sequence segmentation (fig. 4). We then use the simple but effective weighted LZ method described earlier to query token and infer then from the model. We can eventually algorithmically compose with recurrences a sequence containing a succession of voice descriptors such as vowels, fricatives, nasals, transients (fig. 5).

We remain focused on the initial idea of using text-to-speech synthesis for the moment. But, analyzing sounds from token chains could also be used to direct inference and converge toward sound targets [11].

5.3 Long generation using fine-tuning

We observed that the maximum length of audio in Bark was 756 semantic tokens, equivalent to approximately 15 seconds. In contrast, the coarse acoustic model transformer has no such limit. We then recursively feed the latter with sequences of semantic tokens and ensure that we can replicate the same characteristics as for previous iterations. We fine-tune the coarse acoustic model. We had better results fine-tuning all the three models using a parameter we call *history-prompt*. Tuning all three models is also used to target the personality of a specific actor or singer.

We designed a system that dynamically controls tuning over iterations by guiding inferences towards a desired target through controlling probability weights of token, se-

[8] https://github.com/jamesb93/python-flucoma

[9] https://www.flucoma.org

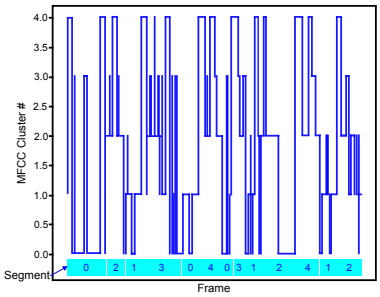

Figure 4. Simple k-mean clustering of MFCC segments before using them as token with our Markovian methods. We have here zoomed into a larger dataset to enhance visibility. Nevertheless, it is evident that the quantity of clusters has a significant impact on the intelligibility of the synthesized voice.

lecting appropriate history-prompts, and manipulating temperature. We use a mixture of text prompting and the acoustic properties method described earlier to automatize long and composed sequences:

1. We iterate our system while having a relatively high temperature in order to widen predictions.
2. When analyzed descriptors reached a specific target, we use the resulting history-prompt and the seed number as fine-tune for subsequent steps.
3. We use seed and history-prompt to generate new versions with low temperature.
4. We gradually increase the temperature step after step until reaching a new target given to the analysis.
5. We repeat from point 2.

Each of the resulting branches can be utilized independently or together to create polyphony. We also employed this interplay to generate hip-hop stems trying to control convergences at specific moments of a song.

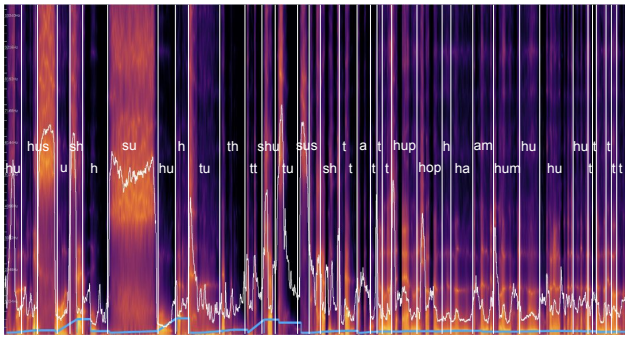

Figure 5. LPC spectrogram showing a generation of `hu`, `tu`, `shu`... Temperature introduced into the coarse acoustic model adds more variety. It also avoids repetition in the sequence by adding more phonemes related to those given by the input tokens. Initially articulated as voice, those segments sound very much like "electro-acoustic music" articulated transitions. A pitch curve in blue and centroid in white have been added over to better visuallize intensity and inflections in this phrase. Notice the end converging toward only one pitch, in blue.

6. FUTURE AND MUSICAL APPLICATIONS

We used these techniques in the production of remixes for popular music singers with the authorization of Warner Music. Those methods are also going to be widely used for a musical English and theatrical French version of John Fosse's play *And We'll Never Be Parted* premiered at T2G National Theatre in September 2025.

The part using GPT-2-like transformers is satisfying for our needs. Despite this, training a large personal model has been difficult if not impossible. We will simplify the workflow using less external libraries and easily port the whole system to FairSeq 2 for example. We want to increase variety and style using Low-Rank Adaptation (LoRA) onto much larger models. We should also be able to merge those models as easily as we do in visual stable diffusion tools.

7. CONCLUSION

The integration of transformer-based TTS synthesis and machine learning into production permits a creative expression between text, sound, literalism and abstraction. We can generate long unique vocal sequences using token engineering controlled by sound-descriptors, and fine-tuning models. This research shows the utility of using simple concepts to achieve intuitive controls. Utilizing transformers might initially seem counterproductive to novelty and creativity. However, integrating parametric processes enables unexpected textual and sonic surprises, distinct from those derived solely from acting. We can thus integrate text and music in our personalized manner.

A Jupyter notebook with all the sequences and audio examples is available here on GitHub.

8. REFERENCES

[1] M. Woo, "L'interprétation musicale des phonèmes, des gestes et des images dans *Machinations* de Georges Aperghis," 2011.

[2] A. Gayraud, R. Mackay, D. Miller, and N. Power, *Dialectic of Pop*, 2019.

[3] "DeepRapper: Neural Rap Generation with Rhyme and Rhythm Modeling," 2021.

[4] N. Obin, "Cries and Whispers - Classification of Vocal Effort in Expressive Speech," 2012.

[5] J. Kammerer, "Unleashing Creativity with Ableton's Beat Repeat: A Comprehensive Guide," 2014.

[6] G. Loizillon, *Diphone Studio*. Ircam, 1999.

[7] J. Bachan, *Efficient Diphone Database Creation for MBROLA, a Multilingual Speech Synthesiser*. Institute of Linguistics, Adam Mickiewicz University, 2010.

[8] O. Lartillot, *OpenMusic LZ 2.2 Library*, 2001.

[9] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," 1978.

[10] P. A. Tremblay, O. Green, G. Roma, and A. Harker, "From Collections to Corpora: Exploring Sounds through Fluid Decomposition," 2019.

[11] B. Hackbarth, N. Schnell, P. Esling, and D. Schwarz, *Composing Morphology: Concatenative Synthesis as an Intuitive Medium for Prescribing Sound in Time*. Contemporary Music Review, Vol. 32, No. 1, 49-59. 2013, 2013.