Specific Music Style Generation

Lukas Elenbergas
Institute of Informatics
Faculty of Mathematics and Informatics
Vilnius, Lietuva
lukas.elenbergas@mif.stud.vu.lt

***Abstract*—In this paper the author tries to generate music resembling specific artists' styles. To do this, a pre-trained MusicGen [3] model (small version) made by the Meta Research team will be fine tuned on the MusicCaps dataset introduced by the Google research team in the article about the MusicLM model [1]. Then, music resembling specific band styles is going to be generated by passing band descriptions generated via the GPT-3.5 Turbo model. The generated sample evaluation is done subjectively.**

***Index Terms*—Music generation, MusicCaps, MusicGen, GPT-3.5**

I. Introduction

Given the recent boon of machine learning models and their capabilities skyrocketing to new heights, the author was interested in exploring music generation models' capabilities of simulating specific sounding music that could emulate particular artists' sounds. A recently publicly released model was selected for this task, namely, the Meta research team's MusicGen model. Due to locally available resources, the scope of this project got limited to a smaller version of the MusicGen model (one with 300M parameters). The model, however, got further trained using the MusicCaps dataset due to it having a wide variety of samples, quite descriptive labels and aspect lists. The GPT-3.5 Turbo model was selected for music artists' description generation for the sake of consistency and due to the availability of this tool to the public.

II. Methods

A. Tools

1) **Pytorch** - machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing.
2) **Audiocraft** [3] - PyTorch library for deep learning research on audio generation.

B. Loss and Optimization

1) As the loss function the *Cross Entropy Loss* was used:

$$-(y\log(p)+(1-y)\log(1-p))-\sum_{c=1}^{M}y_{o,c}\log(p_{o,c}) \tag{1}$$

Where:

a) M - number of classes
b) log - the natural log
c) y - binary indicator (0 or 1) if class label c is the correct classification for observation o
d) p - predicted probability observation o is of class c

2) As the optimizer the *AdamW* was chosen, which is a popular algorithm used in deep learning that helps adjust the parameters of a neural network in real-time to improve its accuracy and speed, but with a modified implementation of weight decay by decoupling it from the gradient update.

C. Models

1) **MusicGen** - a simple and controllable model for music generation. MusicGen is a single stage auto-regressive Transformer model trained over a 32kHz EnCodec tokenizer with 4 codebooks sampled at 50 Hz. Unlike existing methods like MusicLM, MusicGen doesn't require a self-supervised semantic representation, and it generates all 4 codebooks in one pass.

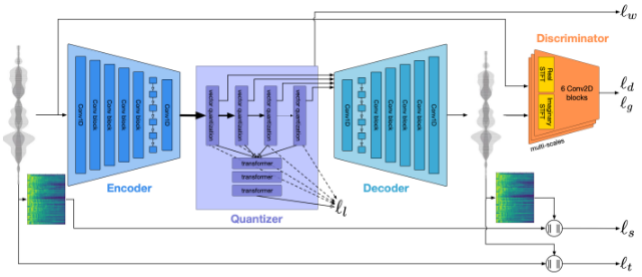

Figure 1 – Visualization of the Meta's MusicGen model.

2) **GPT-3.5 Turbo** - a decoder-only transformer model of deep neural network, which uses attention in place of previous recurrence and convolution-based architectures.

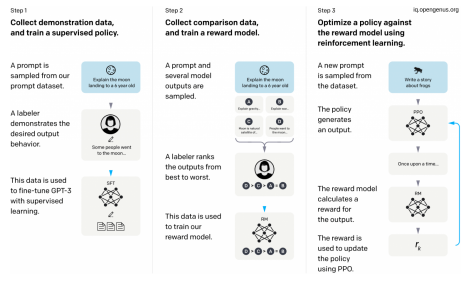

Figure 2 – Visualization of the OpenAI's GPT-3.5 Turbo.

III. Dataset

As the fine tuning dataset, Google's MusicCaps data was used. The MusicCaps dataset contains 5,521 music examples, each of which is labeled with an English aspect list and a free text caption written by musicians. Mostly the aspect lists were used as training labels during the fine tuning of MusicGen. To download and format the dataset GitHub user's *nateraw's (Nathan Raw)* tool "download-musiccaps-dataset" was used [4]. The compilation of the data locally encountered issues due to some of the dataset's videos not being available on YouTube anymore.

IV. Process

After the dataset and all of the necessary libraries to run the MusicGen model were downloaded, it was time to fine tune the model. For this, GitHub user's *chavinlo's* tool "musicgen-tainer" was used [2]. It is based on the Meta team's Audiocraft library and their description of how the training of this model should go. After fine tuning the hyper parameters and training the model on the MusicCaps dataset, it was then used to generate music samples which were supposed to reflect styles of selected music artists.

To generate the descriptions of selected music artists' music, ChatGPT was used to generate the queries. The prompt to generate them was: "Give me a short one sentence description of the music {artist} make without including the band name in the sentence." The model gave pretty descriptive sentences encapsulating selected artists. This description generation was planned to do via code and would have worked perfectly, but the author faced difficulties related to OpenAI monetization model and thus the descriptions were ultimately generated manually.

V. Results

The scope of this paper was limited to subjective evaluation of the generated music samples due to lack of time on the author's part and lack of resources of how objective evaluation of such a project could even be conducted. Nevertheless, the project was somewhat of a success. The samples did have some resemblance of how the original artists could sound, but as always, there was plenty of room for improvement.

The sample size and sample length were relatively small for a complete evaluation. In the future, a survey could be conducted to more properly evaluate such results. Additionally, a more powerful GPU, more training time and more hyper parameter optimization could lead to much better results in the future.

VI. Conclusion

The machine learning world is leaping forward at incredible speeds. The Google research team's MusicCaps dataset and MusicLM model was proposed at the beginning of the year and just half a year later, the Meta research team already came up with an even better somewhat simplified implementation of a music generation model. As things stand right now, the existing models can do impressive work and only time will tell when actual legitimately believable music could be generated that could easily mimic existing artists' works.

References

[1] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[2] chavinlo. musicgen-trainer. https://github.com/chavinlo/musicgen_trainer, 2023.

[3] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

[4] Nathan Raw. download-musiccaps-dataset. https://github.com/nateraw/download-musiccaps-dataset, 2023.