Text Tune : Text to Music Generation

Chethana Muppalam
chetha@pdx.edu

Vaishnavi Srinath
vsrinath@pdx.edu

Abstract

Generative models have demonstrated remarkable abilities in generating text, images, and video content based on prompts. However, audio generation using prompts remains relatively under-explored. This study introduces Text-Tune, an autoregressive audio generation model capable of producing music from textual descriptions based on a custom theme/music style and duration. Text Tune is developed on top of MusicGen pretrained model. The primary focus of this research is to analyze existing audio generation models and conduct experiments with Meta's AudioCraft Library's MusicGen models, incorporating prompt engineering.

We employed a zero-shot prompting strategy using GPT-3.5 and compared the outcomes of the generated audio with the existing MusicGen base models. All audio samples generated by Text-Tune were rendered at a sampling rate of 32kHz. To assess the quality of the generated music, we evaluated it using FAD and SNR scores on the MusicCaps Dataset. Access to music samples, code, and models is provided on [GitHub Link].

1 Introduction

In recent times, generative music has witnessed remarkable advancements, particularly highlighted by the emergence of groundbreaking models like MusicGen developed by Meta, MusicLM by Google, and most notably, Lyria by DeepMind. Utilizing techniques such as diffusion-based encoding/decoding of audio signals and a transformer-based language model, these models are capable of generating music from text, mixing and matching genres/melodies, and creating music of lengths several factors longer than their trained samples.

Even with the state-of-the-art models, there are a few shared limitations. Primarily, the majority of them specialize in generating instrumental music exclusively. They heavily rely on textual prompts specifying genre, rhythm, tempo, and mood. In addition, we observed that these models lack specificity or nuance in their outputs. Many of the prompts used in Meta's AudioCraft examples are very basic and contain vague genre/instrument descriptions, such as "Classic reggae track with an electronic guitar solo." The quality of music generated and the noises in the generated audio were variegated and better in stereo models than in base models. Our research attempts to refine these models to cater to specific moods or occasions. We aim to assess how well the generated music aligns with the intended theme provided by the user. Recognizing the diversity in musical styles, moods, and instrumentation across different themes, we utilize the GPT-3 model to engineer prompts that grasp the nuances of the user's requests effectively.

2 Related Work

The development of MusicGen at Meta draws inspiration from several key advancements. MusicGen tackles conditional music generation, where music is created based on instructions. Unlike prior methods that use multiple stages or increase detail in stages, MusicGen utilizes a single transformer model for efficiency. It works with compressed musical tokens instead of raw audio, allowing control through text descriptions or melodies. Compared to previous models, it generates high-quality music, offering greater control through user input with a simpler architecture.

MusicGen heavily relies on the principles established by Encodec. Encodec, a state-of-the-art neural audio compression algorithm renowned for its high-fidelity reconstruction capabilities[High Fidelity Neural Audio Compression]. Encodec leverages a powerful architecture called an autoencoder. This structure allows it to efficiently learn compressed representations of audio data. However, Encodec takes this a step further by incorporating a technique called residual vector quantization (RVQ) within a bottleneck layer. This bottleneck

acts like a compression point, generating multiple streams of audio tokens. Each stream captures a specific aspect of the original audio information, from high-level details to finer nuances. This innovative approach empowers Encodec to achieve remarkably effective reconstruction of the original audio. The design of MusicGen heavily relies on the compression principles established by Encodec. Coming to the Architecture, MusicGen utilizes an autoregressive single-stage transformer architecture. Prior work in music generation with transformers includes models like Jukebox from OpenAI, which demonstrated impressive music generation capabilities but with higher computational demands.

Before the emergence of Meta's MusicGen, another model called AudioGen [AudioGen: Textually Guided Audio Generation] laid down crucial groundwork for conditional music generation. [2] introduced a significant concept in this domain by suggesting the modeling of multiple streams of speech tokens in parallel, employing a delay approach where streams include an offset. This ingenious technique, also adopted in MusicGen, proved instrumental in advancing the field.

Other related works include similar text-to-music models from [8], with the key difference being that they use hierarchical modeling of audio representations. At its core, MusicLM leans more heavily on natural language processing (NLP) techniques to interpret and analyze text inputs, understanding the nuances of the descriptions provided by using a hierearchy of autoregressive models.

Another generative music model released after MusicGen, Lyria by Google DeepMind. Lyria is a groundbreaking AI model that pushes the boundaries of music creation. Unlike previous models, Lyria excels at generating high-fidelity music, including both instrumental compositions and vocals. This innovation promises to empower users with more control over the style and direction of the generated music. Notably, Lyria's capabilities extend beyond pure generation. It can also transform existing pieces or seamlessly continue music in a chosen style. This paved way foe exciting applications like Dream Track on YouTube, where creators can craft short music clips using the voices and styles of popular artists.

3 Methodology

3.1 Dataset

The research study uses the MusicCaps Dataset to evaluate the quality of audio generated by the text-to-music model. MusicCaps is a collection of music samples paired with descriptive text captions. It's designed to aid research in tasks involving music and natural language processing. The dataset contains over 5,500 snippets, each lasting 10 seconds.There are two main ways the music is described in the dataset. One is a list of musical aspects, categorizing things like genre, mood, instruments, and vocal characteristics. The other is a free-form caption written by musicians, offering a more subjective and detailed description of the music. For instance, the caption might mention the feeling the music evokes or specific details about the melody or rhythm. We used the dataset available on Kaggle in a CSV format containing information for each music sample. This information doesn't include the audio file but includes the start and end time of the clip within a YouTube video (most samples come from YouTube), labels from a large audio classification dataset called AudioSet, the musical aspect list, the free-form caption, and some additional identifiers.

3.2 Pretraining

The pretrained MusicGen model consists of a pretrained text encoder (T5), a transformer-based decoder language model (LM), and a neural audio codec autoencoder known as Encodec, which learns a discrete representation of audio.

This pretrained model was originally trained from scratch on 20K hours of licensed music and involves another model which Meta calls Encodec, which was separately trained as an autoencoder for audio waveforms, consisting of an encoder, a quantizer, and a decoder. Using residual vector quantization (RVQ) along with a novel method for interleaving audio tokens, the Encodec encoder can produce a compressed representation of parallel streams of audio tokens over time across various codebooks. This, along with the prompt embedding obtained through a forward pass of a pretrained text encoder, is passed as input into the decoder LM, which produces a sequence of audio tokens. These tokens are then passed into the Encodec decoder, which finally generates the output audio waveform.

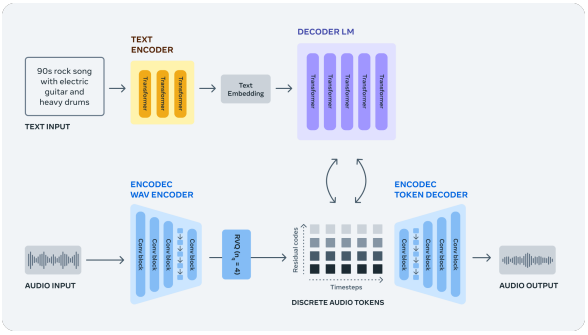

Figure 1: MusicGen Architecture

3.3 Prompt Engineering

This study leverages prompt engineering with GPT-3.5 Turbo API to achieve finer control and richer nuance in music generation. Existing models often struggle with a lack of user control and tend to routine instrumental music. To address this, we implemented a zero-shot prompting strategy. This approach involves crafting prompts that capture the user's desired theme, mood, and potentially specific stylistic elements. However, unlike traditional prompting, Text-Tune utilizes GPT-3.5 Turbo's ability to explore possibilities based on the user's input. This empowers Text-Tune to go beyond simply capturing the keywords and delve deeper, crafting prompts that incorporate the user's expectations, emotional intent, and potentially specific stylistic preferences. These enriched prompts are then fed into the pre-trained MusicGen model, guiding the generation process towards music that more closely aligns with the user's artistic vision. We conduct experiments to evaluate if this approach will significantly improve upon the limitations of basic genre/instrument prompts and lead to music that resonates more deeply with the user's intended theme and mood.

3.4 Generation

The music generation process is fairly straight forward. The enriched prompt from GPT-3, along with the desired duration, are then fed into the MusicGen model. MusicGen employs a text encoder, to convert the enriched prompt into a numerical representation. This representation captures the essence of the user's desired music.

Next, MusicGen's decoder (LM) trained for audio generation, comes into play. This decoder operates in an autoregressive manner, meaning it predicts the next audio token (a tiny slice of audio information) based on the previously generated tokens and the encoded prompt information. This iterative process continues until the desired music duration is achieved.

Finally, a decoder-encoder architecture within MusicGen takes the generated sequence of audio tokens and converts it into an actual waveform. This waveform represents the sound waves the user will hear when the music is played back.

Through this two-step process, Text-Tune aims to bridge the gap between human imagination and musical creation. By leveraging enriched prompts and a powerful music generation model.

4 Experiments

We ran all the experiments on a Google Colaboratory Platform utilizing Nvidia V100, A100.

4.1 Experimental Setup

4.1.1 Pre Trained models under experimentation:

- MusicGenSmall-300m
- MusicGenLarge-3.3b
- MusicGenStereoSmall - 1.5b
- MusicGenStereoLarge - 3.3b

4.1.2 Experiments Performed:

- Music Generation using MusicGenSmall without Prompt Engineering - (**Model 0**)
- Music Generation using MusicGenLarge without Prompt Engineering - **(Model 1)**
- Music Generation using MusicGenSmall without Prompt Engineering - **(Model 2)**
- Music Generation using MusicGenLarge with Prompt Engineering **(Model 3)**

4.2 Metrics

We analyze various metrics to assess Text-Tune, focusing on two key aspects: the audio's quality and its fidelity to the text description.To comprehensively assess the effectiveness of our method, we leverage a two-pronged evaluation strategy utilizing both **Frechet Audio Distance (FAD)** and **Signal to Noise Ratio (SNR)**.

FAD, drawing inspiration from image quality assessment advancements, stands out as a

reference-free metric. This means it can assess the overall perceptual similarity between the generated audio and authentic, high-quality audio without requiring a perfect reference for comparison. In essence, FAD gauges how well the model captures the natural characteristics of desired audio, even in the absence of a pristine example. This is particularly valuable when dealing with creative audio generation where a single "correct" version may not exist. Generally, a higher FAD score indicates a higher quality recording with minimal or no glitches.

SNR offers a more direct approach, measuring the strength of the desired audio signal relative to any background noise. By analyzing both FAD and SNR, we gain a richer understanding of how effectively our generated audio replicates high-fidelity audio. We can assess not only how closely the generated audio resembles desired qualities but also how effectively it minimizes unwanted noise artifacts. This combined evaluation approach provides a more comprehensive picture of the method's strengths and weaknesses, allowing for further refinement and optimization. A higher SNR indicates a stronger audio signal relative to background noise.

Qualitative evaluation is central to assessing text-to-music generation, as it tackles the subjective nature of music. Unlike metrics that measure technical aspects like pitch accuracy, qualitative evaluation focuses on how human listeners perceive the music. This can involve user studies where participants rate the generated music on aspects like coherence, emotional response, or alignment with the textual prompt. While subjective and requiring careful design, qualitative evaluation offers valuable insights into how well the model captures the essence of music and evokes emotions in listeners, which is ultimately the goal of music creation.

4.3 Results

We start by presenting results, Table 1 and Table 2, of the proposed method on the task of text-to-music generation.

4.3.1 Quantitative Analysis

The results show significant improvements in the FAD metric when using GPT-3 with both musicgen models, potentially due to its strength in understanding and interpreting text prompts. This could be because GPT-3 helps the music generation process by providing a more focused and coherent musical direction. This shows that text-to-music generation, leveraging external information sources during the creation process, can further enhance audio quality. In the musicgen-small model, the FAD metric decreased from 1.182 to 1.057, indicating a substantial improvement in audio smoothness. This might be attributed to GPT-3's ability to guide the music generation towards a more consistent and unified style. However, the difference between the models remains minimal, especially with GPT3.

Table 1: musicgen-small

Metric	Base	Text-Tune
FAD	1.053	1.094
SNR	-6.64	-3.8

Table 2: musicgen-large

Metric	Base	Text-Tune
FAD	1.182	1.057
SNR	-2.3	-5.15

SNR scores are generally higher for musicgen-large in the base configuration, but lower with Text Tune. This suggests that musicgen-large might introduce more noise when used with Prompt Engineering, potentially due to an overload of information or conflicting musical styles suggested by the text prompt and retrieved samples. While the SNR improved in the musicgen-small model from -2.3 to -5.15, indicating better noise reduction, the musicgen-large model showed an increase in SNR from -6.64 to -3.8, suggesting a slight degradation in noise reduction performance. Further investigation is needed to determine if this is due to the model itself or the interaction with Prompt Engineering.

4.3.2 Qualitative Analysis

Our qualitative analysis of Music Stereo Large and Music Stereo Small models provides encouraging evidence. The generated audio exhibits improvements in noise reduction and overall clarity. This suggests a potential correlation between model size and noise handling capabilities. Models with a higher number of parameters (like the 1.5b and 3.3b parameter Music Stereo models) might be par-

ticularly adept at mitigating noise in the generation process.

5 Conclusions

In conclusion, our experiments explored the potential of leveraging GPT-3.5 for prompt engineering in Meta's AudioCraft MusicGen models. The findings demonstrate the effectiveness of this approach in enhancing specific aspects of audio generation. Notably, FAD scores improved significantly with both MusicGen models, suggesting a positive impact on audio smoothness and potentially, musical coherence. This supports the notion that text-to-music generation can benefit from external information sources like GPT-3.5 to guide the creative process. However, the impact on SNR metrics was mixed. While the smaller MusicGen model exhibited improved noise reduction, the larger model showed a slight increase in noise. Further investigation is warranted to understand if this is due to inherent model limitations or the interplay between GPT-3.5 and the larger model's capacity.

6 Insights

We employed human evaluation for assessing the MusicGen Stereo models. The quantitative metrics of the MusicGen Stereo models were not evaluated due to their high computational demands, such as requiring GPUs, additional RAM, and significant processing time.

While the MusicCaps dataset offers a readily available resource for audio generation research, its limitations in audio quality hinder its effectiveness as a training dataset for high-fidelity music generation tasks. There is a need of high-quality benchmark training data for achieving optimal audio generation results.

Furthermore, the licensing restrictions associated with commercially licensed datasets like Shutterstock and Pond5 prevent their use for pre-training our own models. This limitation is a common challenge faced by researchers in the field, and are exploring open-source alternatives for pre-training audio generation models.

7 Future Work

To further enhance the capabilities of this music generation system, we envision several avenues for future exploration:

Model Fine-Tuning with Popular Music: We plan to fine-tune the model on a dataset comprised of the most popular album songs. This exposure to commercially successful music will equip the model to generate content that resonates with a broad audience.

Remix Composition: The model will be trained to compose remixes, fostering creativity and allowing users to explore variations of existing pieces.

Multimodal Exploration: We aim to enable multimodal exploration, empowering users to combine musical prompts with other forms of data like images or videos for a richer creative experience.

Multi-Lingual Support: Future development will focus on incorporating multi-lingual support, allowing users to provide prompts and receive generated music in various languages, fostering global accessibility and artistic expression.

8 Ethical Considerations

Text-Tune raises several ethical considerations regarding data, bias,and environmental impact.

Data: The Foundation of Fairness

The MusicCaps dataset, the bedrock of Text-Tune, deserves scrutiny. Was it built with inclusivity in mind, or does it favor specific genres or cultures? A narrow data pool can lead to biased models that struggle with diverse musical styles, limiting creativity and potentially excluding users. Additionally, the quality and objectivity of the captions within the dataset are crucial. Inaccurate or subjective labels can lead Text-Tune astray, generating music that misses the user's intent.

Avoiding Bias: Fostering Fairness

Musical style bias can be a real concern. If Text-Tune is heavily trained on a particular genre, it may struggle to generate music outside that comfort zone. This could stifle creative expression and leave users interested in under-represented styles out in the cold. Furthermore, cultural biases within the data could lead to the generation of music that reinforces stereotypes or lacks cultural

sensitivity.

Minimizing the Footprint: Environmental Responsibility

Training and running large language models like Text-Tune requires significant computing power, which translates to a hefty energy consumption. We must consider the environmental impact and explore ways to make Text-Tune more energy-efficient.

Building a Sustainable Future for Music

Transparency regarding the MusicCaps dataset is crucial. Researchers should openly share its origin and composition to allow for bias assessment. Efforts to expand the dataset with a wider variety of musical styles and cultural perspectives are essential. Regular audits of Text-Tune for potential bias and implementing mitigation strategies are vital. Educating users about Text-Tune's capabilities and limitations is key to preventing misuse. Finally, exploring ways to train and run Text-Tune using more energy-efficient platforms or renewable energy sources is a step towards a sustainable future for music.

By addressing these ethical considerations, we can ensure Text-Tune reaches its full potential as a tool for creative expression, fostering a diverse and inclusive musical landscape while minimizing potential harm.

9 References

[1] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438, 2022.*

[2] Felix Kreuk, Jade Copet, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, Alexandre Défossez. Simple and Controllable Music Generation. *arXiv preprint arXiv:2306.05284, 2023*

[3] Junho Park, Sanghoon Kim, Jaemin Kim, Eunkwang Jang, Jaeyoung Jeong, Juyong Kong, Junyoung Nam, Namgyu Kim. AudioGen: Textually Guided Audio Generation. *Proceedings of the 2022 Conference on Audio Processing and Speech Recognition (APSR), pages 224-228, 2022*

[4] Hany Farid, Rohan Rao, Adam Roberts, Michael Schuster, Joshua Batson, David Courville. MusicLM: Generating Music From Text. *arXiv preprint arXiv:2301.11325, 2023*

[5]Siyuan Luo, Yiqin Tan. LCM-LoRA: A Universal Stable-Diffusion Acceleration Module. *arXiv preprint arXiv:2311.05556, 2023*

[6] zalea Gui, Mengyu Qian, Kejun Wang, Yuxuan Chen, Zhe Gan, Bojan Radicioni, Li Zhang. ADAPTING FRECHET AUDIO DISTANCE FOR GENERATIVE MUSIC EVALUATION. *arXiv preprint arXiv:2311.01616, 2023*

[7] Junho Park, Sanghoon Kim, Jaemin Kim, Eunkwang Jang, Jaeyoung Jeong, Juyong Kong, Junyoung Nam, Namgyu Kim. CLAP: Learning Audio Concepts From Natural Language Supervision. *arXiv preprint arXiv:2206.04769, 2022*

[8] Michael Xue, Edward Diller. Artist-Specific Finetuning for Generative Music with LoRA and Textual Inversion

10 Group Contribution Statement

We agree that all group members made a valuable contribution and therefore believe it is fair that each member receive the same grade for the discussion.