WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Research Paper on AI Music Generator Using Streamlit and MusicGen

Saket Patil[1], Vaishnavi Patil[2], Dnyaneshwari Navathar[3], Aditi Ghugarkar[4], Prof. Sheetal Nirve[5]

[1,2,3]Department of Computer Engineering, K J College of Engineering & Management Research Pune, India
[4]Prof., K J College of Engineering &Management Research, Pune, India

ABSTRACT

In this paper, we present a novel implementation of a text-to-music generation system utilizing pre-trained AI models for conditional music generation. Our system leverages the facebook/musicgen-small model, a transformer-based model, to generate music based on textual descriptions provided by users. The application is deployed on Streamlit, a Python-based platform for building interactive web applications, which allows seamless real-time interaction and rapid prototyping. The goal of this research is to explore the capabilities of AI in the creative domain of music generation and provide an accessible tool for users to generate music based on their inputs. This paper discusses the architecture, deployment strategy, model performance, and future directions in the development of text-to-music generation systems.

Keywords : musicgen-small, streamlit, text-to-music generation

INTRODUCTION

Music generation is a complex task that requires understanding various aspects of music, including melody, rhythm, harmony, and timbre. Historically, music has been created by human composers, but recent advancements in machine learning and artificial intelligence (AI) have introduced automated systems capable of generating music. AI-based music generation systems can assist composers, create soundtracks for video games or movies, and even produce unique music pieces based on user input. With the rise of deep learning models and transformer architectures, music generation from non-musical inputs, such as text descriptions, has become a reality.

METHODODLGY

Model Selection

The facebook/musicgen-small model was chosen for this project due to its balance between model size and performance. It is a pre-trained generative model designed to produce short musical sequences from text prompts. The small variant of the model was selected to ensure faster inference times while maintaining the quality of generated music.

Model Architecture

MusicGen is a transformer-based model trained on large datasets of music. The transformer architecture is well-suited for sequence generation tasks like music generation because it can capture long-range dependencies in data. The model takes a text description as input and **generates a sequence of audio tokens, which are then converted into audio.**

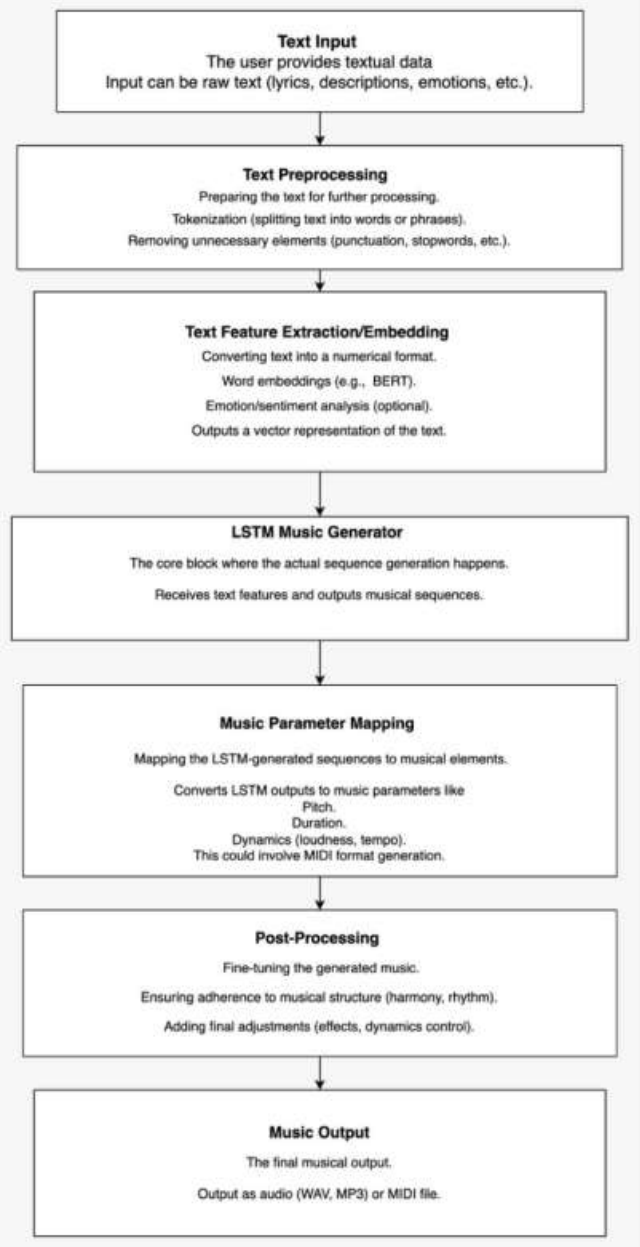

Fig: Process flow of text to music generation

Application Development

Streamlit for Frontend Development

Streamlit was chosen as the deployment platform due to its simplicity and effectiveness in building interactive applications. Streamlit allows for the seamless integration of sliders, text inputs, and buttons without the need for traditional HTML, CSS, or JavaScript. This ensures a quick development cycle and a user-friendly interface for generating music.

System Workflow

1. User Input: The user provides a text description of the music they want to generate (e.g., "a calming piano melody").
2. Music Generation: The description is processed using the AutoProcessor and passed to the MusicGen model for conditional music generation.
3. Audio Processing: The generated audio tokens are converted into a waveform and normalized for output.
4. Audio Playback and Download: The system outputs the music as an audio player in the web interface, allowing users to listen to the generated track. Users also have the option to download the audio file in .wav format.

Code Implementation

The system consists of the following key components:

- Loading the Model: The facebook/musicgen-small model is loaded using AutoProcessor and MusicgenForConditionalGeneration classes.
- Generating Music: User input is fed into the model, and the generated audio tokens are converted to audio format.
- Audio Output and Download: The generated audio is played back in Streamlit and made available for download.

Results and Evaluation

User Interaction and Feedback

Initial user tests show that the application successfully generates music from text descriptions. Users found the interface easy to use, and the real-time feedback through the Streamlit interface allowed for smooth interaction.

Model Performance

The facebook/musicgen-small model demonstrated robust performance in generating music conditioned on diverse text prompts. Short prompts (1-2 words) resulted in basic musical compositions, while longer, more descriptive prompts allowed for richer and more complex music.

Limitations

Despite its success, the model has limitations, including:

- Length Constraints: The maximum length of generated music is limited by the model's capacity.
- Lack of Fine Control: The generated music may not always perfectly match user expectations, especially for highly specific or abstract prompts.
- Inference Speed: While the small variant of MusicGen is faster than larger models, generating music still requires a few seconds per sample, which could be improved.

Conclusion

The text-to-music generation system developed in this project demonstrates the potential of AI in the creative domain of music composition. Using transformer-based models such as facebook/musicgen-small, users can generate music from simple text descriptions in real-time through an intuitive web interface. Streamlit was instrumental in enabling rapid development and deployment, offering an accessible platform for non-experts to interact with complex AI models.

Future Work

Future work could include:

- Expanding the Model's Capabilities: Integrating larger models for higher-quality music generation or fine-tuning existing models for more accurate genre-specific music.
- Adding Fine-Grained Controls: Allowing users to specify parameters such as tempo, instruments, or mood, giving them more control over the generated music.
- Improving Scalability: Deploying the system on more robust cloud infrastructures to handle higher traffic and faster music generation.

References

- Vaswani, A., et al. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems*.
- Dhariwal, P., & Nichol, A. (2020). Jukebox: A Generative Model for Music. *arXiv preprint* arXiv:2005.00341.
- Streamlit. (2023). Streamlit Documentation. Available at: https://docs.streamlit.io/

S

ome evaluations:

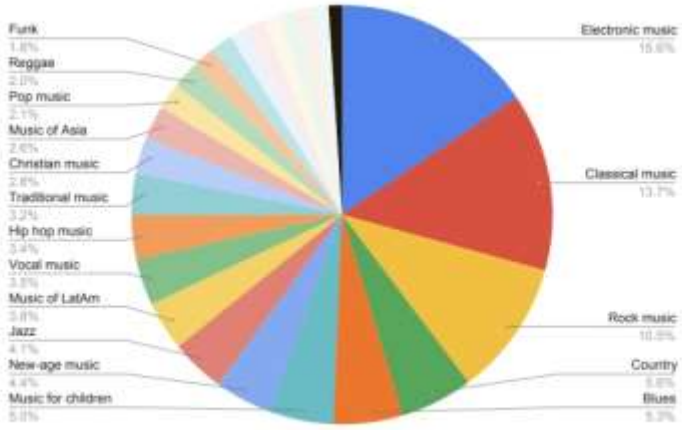

Fig. 1 – Genre contribution in model

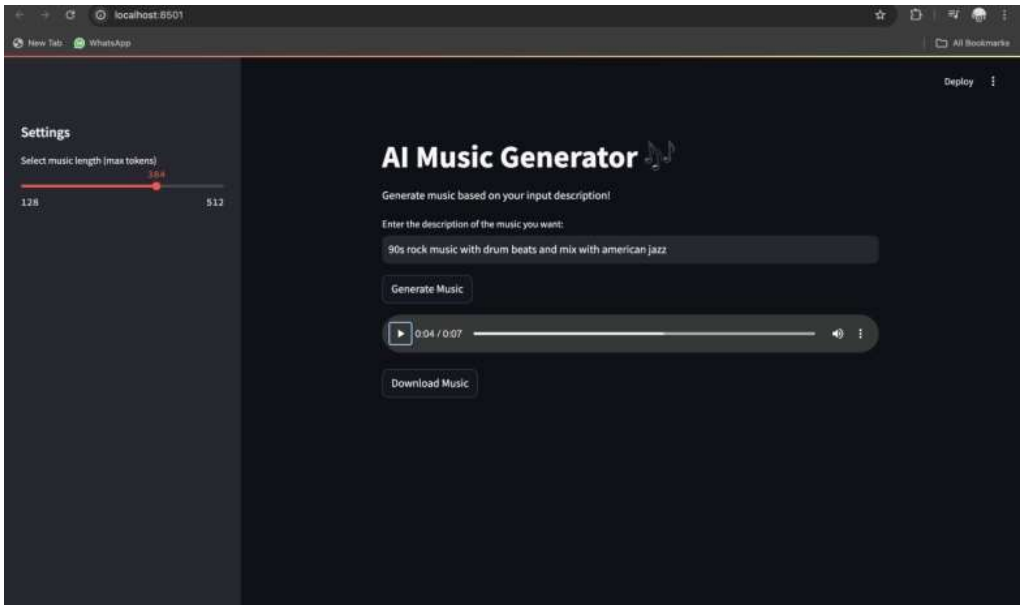

Fig. 2: local deployment using streamlit and user interface for human listener study