Rhythm Composer: Music Composer

Saurav Kumar[1], A Durga Madhab Patro[2], Raushan Raj[3], Dipanshu Kumar Mahato[4], Dr. Suprava Devi[5], Dr. Barnali Sahu[6]

Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar, Odisha, India
sauravmishra623@gmail.com, durgapatro2002@gmail.com, rraushan9247@gmail.com, dipanshumahato@gmail.com, supravadevi@soa.ac.in, barnalisahu@soa.ac.in

Abstract.
The Rhythm Composer project presents an innovative approach to automated music composition by integrating artificial intelligence with audio processing techniques. This system simplifies the traditionally complex process of rhythm-based music creation by combining user-generated or AI-generated lyrics with synthesized vocals and beat patterns. Utilizing Python, GPT-Neo for lyrical generation, TTS for vocal synthesis, and Bark model for music generation, the platform offers a modular and scalable architecture suitable for both educational and creative applications. The system provides a seamless pipeline from text input to fully composed audio output, enabling users with little to no musical training to produce rhythmically coherent music. Designed for flexibility, it supports experimentation across genres and serves as a practical tool in fields like music education, game development, therapy, and solo production. The project underscores the potential of AI to democratize music creation and bridges the gap between computational power and artistic expression.

Keywords: AI Music Generation, Text-to-Music, Bark Model, Lyric-to-Audio Conversion, Transformer Models

1 Introduction

The Rhythm Composer is an innovative system automating rhythm-based music creation by combining pre-written lyrics, synthesized vocals, and beat patterns into cohesive audio tracks. It streamlines composition using audio synthesis, modular design, and interactive programming, mimicking human creativity to produce structured, harmonized music with minimal manual effort. This flexible tool serves musicians, developers, and hobbyists, simplifying rhythm track production while allowing for customization and scalability. With applications in music education, independent music production, game development, and music therapy, its modular architecture ensures efficient development. The Rhythm Composer uses computer technology to help people create music, even if they do not have much musical knowledge. It shows how technology can make music-making easier, faster, and available to everyone, compared to traditional ways.

1.1 Motivation(s)

The Rhythm Composer project makes it easier to create music by using technology to automatically generate beats and vocals. It removes the need for deep musical skills or expensive tools. The main goals are to help people without formal music training, save time by cutting down on manual work, teach rhythm in a fun way, and encourage creativity. It can be used in many areas like education, games, media, therapy, and making music alone. In the end, the project wants to make music creation open to everyone by using AI and sound tools in a simple and easy to use platform without needing costly software or a studio.

1.2 Objectives(s)

This project plans to build an AI system that can automatically create rhythm-based music from text It will turn lyrics either written by the user or made by AI into singing using text to speech tools Then it will match those vocals with beats to produce full songs. The goal is to simplify, accelerate, and democratize music composition for individuals regardless of their musical experience. This approach seeks to lower the entry barrier to music production, fostering creativity and enabling broader participation in the musical arts.

1.3 Original Contributions

This work presents a unique integration of AI technologies, including GPT-Neo for lyrical generation, Text-to-Speech (TTS) for voice synthesis, and the Bark model for music generation, all within a single, cohesive pipeline. A key contribution is the system's modular design, enabling individual components to be modified or scaled independently. Furthermore, it offers real-time user interaction via a FastAPI interface. Unlike most existing tools that specialize in a single task, this system handles the entire process from lyric-to-audio generation in one streamlined operation, a capability rarely found in current solutions. This comprehensive approach significantly simplifies the music creation workflow and broadens accessibility for users.

1.4 Paper Layout

This report is split into logical sections: starting with introduction and motivations, followed by literature review, then the architecture, methodology, and technologies used. Results and outputs of the system are then discussed, including evaluation methods and screenshots. Finally, the paper wraps up with future improvements, references, and appendices.

2 Literature Survey

In the literature, many AI music generation models were considered, and Jukebox and MusicLM are not exceptions. But such systems tended to have limitations in terms of lacking fine-grained control or high computing requirements. Although Bark by Suno.ai did provide better generation speeds, it sometimes failed on providing natural transitions on its output. After a rather thorough scanning of the whole spectrum of available solutions, we noted one clear roadblock that could not be bridged; the lack of a tightly-packaged system with the ability to integrate all the key components of music: lyrics, vocals, and beats in one and the same pipeline. To overcome this shortcoming, our system would have modular but tightly combined components that would together produce their coherent musical compositions in response to textual input. It is efficient and of high quality but also very interactive and a user can customize genre, mood and structure in real-time. In contrast to the earlier models which aimed to address an independent part of music production, our work guarantees a harmonized combination of lyrics, vocal treatment, and instrumentation. Such connectivity allows it to create a more natural and user-friendly experience of creativity, among both the casual enthusiasts and professional musicians.

3 Proposed System/Model

The Rhythm Composer system is designed to follow a clear two-step process which includes generating lyrics and then creating the voice output Users have the option to either enter their own lyrics or let the GPT Neo model generate them automatically using artificial intelligence Once the lyrics are ready they are converted into vocal audio using text to speech technology A modular backend powered by FastAPI takes care of managing this entire process smoothly This design supports fast performance interactive features and makes the music creation process simple flexible and efficient for users of all skill levels.

3.1 Methodologies Used

We leveraged Python for overall system orchestration. The GPT-Neo model is employed to generate lyrics based on provided themes or prompts. Subsequently, a Text-to-Speech (TTS) engine converts these generated lyrics into vocal audio clips. The audio merging and synchronization are carefully handled to ensure a cohesive output. The system's architecture is intentionally broken down into modular components to enhance flexibility and maintainability. For a lightweight yet responsive web interface, we utilized FastAPI. This combination of technologies allows for efficient processing and a user-friendly experience in the music generation pipeline.

3.2 Schematic Layout of the proposed system/model

The system generates AI music through a dual-path process, starting with a user's text prompt. This prompt initiates two parallel streams: one for lyric creation and the other for music audio production. In the first stream, a GPT Neo Model generates lyrics, outputting them as a text file. Simultaneously, a Music Generation model, such as Bark, produces musical audio in the second stream, resulting in an MP3 file. This architecture enables the simultaneous creation of both textual lyrics and corresponding music from a single user input. Schematic layout for "RHYTHM COMPOSER: MUSIC GENERATOR" is shown in **Figure 1.**

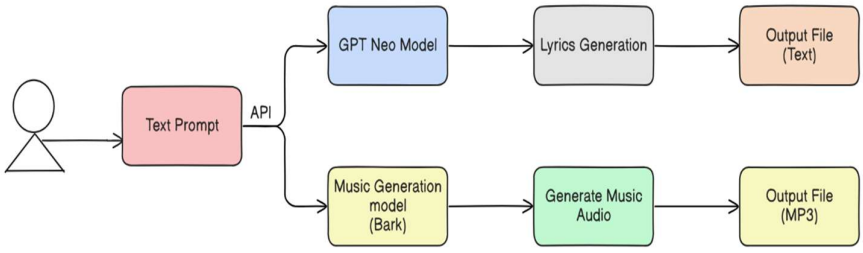


Figure 1. Schematic Layout for the Rhythm Composer

3.3 System Requirements

To run this system effectively, users will need specific hardware and software configurations. Recommended hardware includes an Intel i7 or Ryzen 7 CPU, an NVIDIA RTX 3060 GPU or better, at least 16 GB of RAM, and a minimum of 512 GB SSD storage. On the software front, the system is compatible with Windows 10/11, Ubuntu 20.04+, or macOS operating systems. It requires Python 3.10+ and several key packages such as transformers, torch, FastAPI, and uvicorn. Development and testing can be facilitated using tools like VS Code, Postman, and Git.

3.4 Proposed Algorithm(s)

The Rhythm Composer system's core functionality relies on a blend of advanced algorithms and technologies. Python serves as the primary programming language, orchestrating the entire process. At the heart of text generation is the GPT-Neo model, leveraging the Transformers Architecture for enhanced text processing and self-attention capabilities. This facilitates the Text-to-Music Generation, converting initial text prompts into both lyrics and musical elements. A crucial aspect is Music Structuring, which intelligently aligns the generated lyrics with musical patterns. The system incorporates various Composition Approaches and a Neural Architecture to ensure that the AI-generated music adheres to established composition rules, resulting in structured and coherent musical outputs.

4 Experimentation and Model Evaluation

The thorough experimentation and subsequent assessment undertaken determined the Rhythm Composer system's effectiveness and capabilities. Our evaluation methodology centered on crucial elements like the musical harmony and realistic quality of the generated pieces, the precision of vocal synchronization with rhythmic structures, and the consistency with the initial text inputs. We utilized both subjective evaluations, involving human perception to judge musicality and aesthetic appeal, alongside objective measurements where applicable. The central aim of these assessments was to confirm the system's capacity for efficiently creating high-quality, rhythm-driven music, concurrently pinpointing areas for future enhancement and optimization to improve user experience and musical authenticity.

4.1 Depiction Results

The main points brought out are the clarity of the vocal synthesized and the naturalness, the accurate combination with the different rhythmic patterns, and the general musical integrity of the pieces. These illustrations confirm the main purpose of the project, which involves automating the generation of music and gives tangible samples of how integrated AI models turn out to effectively generate structured and harmonic pieces of music without much human interference. The results presented on the show reveal the prospect to enable users to generate mu-sic with little or no technical or music proficiency that the system has. Moreover, the flexibility of the system in genres and style implies that the system is very versatile, serving a broad applications spectrum. This makes the platform a great experimenting tool as well as a serious music production tool.

4.2 Validation/System Performance Evaluation

A preliminary review aimed at assessing the quality of the produced outputs was also done which included assessing the relevancy, tone and structural soundness. Although formal quantitative measures are not yet used, the initial testing confirmed the possibility of the model to generate various types and contextually-suitable lyrics. Inception of this evaluation underscored the ingenuity of the system in the generation of creative contents. In future versions, it is intended to introduce higher-order assessing methods, which include automatic scoring systems and extended user satisfaction questionnaires to make the results of the performance more stringent and statistically proved. Besides this, there are resources being used to compare the system with the existing models in controlled environments where a comparison of the performance of the two can be made. Those in the future may have domain specialists as part of future assessments to add musicality, coherence, and emotional resonance to be even more refined.

4.3 Discussions on Contributions

Our system addresses a significant void in current AI music tools by providing real-time lyric-to-audio generation within a single, integrated package. Each of its modular components operates independently yet connects seamlessly, ensuring efficient data flow. Unlike existing solutions that often necessitate cumbersome manual export and import between different processing stages, our intuitive interface consolidates the entire workflow within a unified application. This easy-to-use system makes AI music creation possible for people who do not have much technical or musical knowledge. As a result, it gives users more creative freedom and helps them try out new music ideas more quickly.

The user interface of our AI music generator is shown in **Figure 2**. User provides the prompt in the box. After clicking the generate lyrics button, lyrics is generated. The generated lyrics are then displayed in the output area below the prompt box. Users can clear the input or regenerate new lyrics based on different prompts, making the interface simple and interactive.

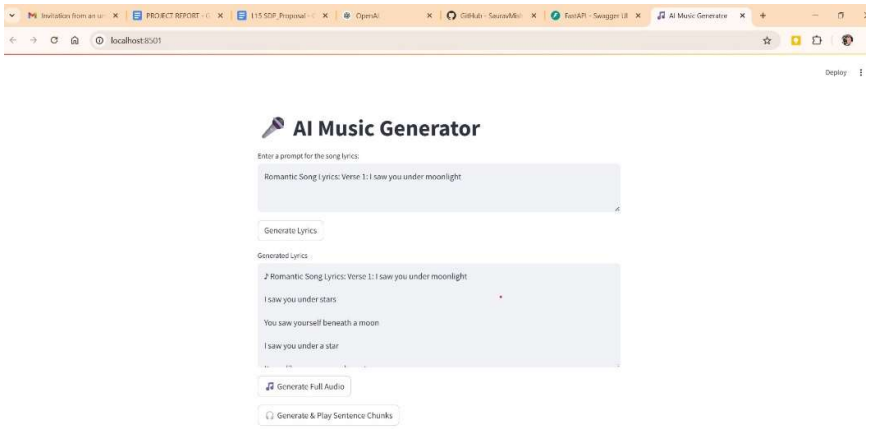


Figure 2. User Interface of the AI Music Generator with prompt

After clicking the Generate Lyrics button, a response body containing the generated lyrics is returned. This response is based on the prompt provided by the user. The structure and content of the response body for a sample prompt is illustrated in **Figure 3**. It includes metadata such as generation time, model used, and the generated text content.

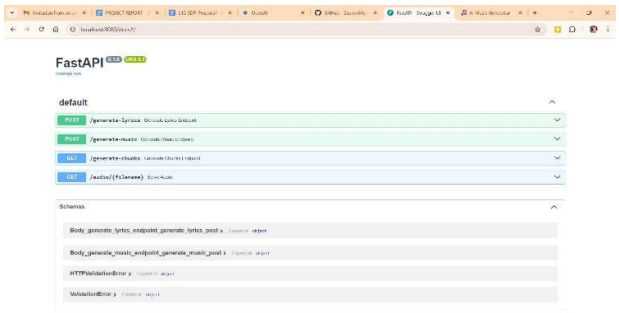


Figure 3: API End Points for Rhythm Composer

The generated lyrics and metadata are stored as key-value pairs in a JSON file for record-keeping. As shown in **Figure 4**, this structured format enables easy parsing and integration with other systems.

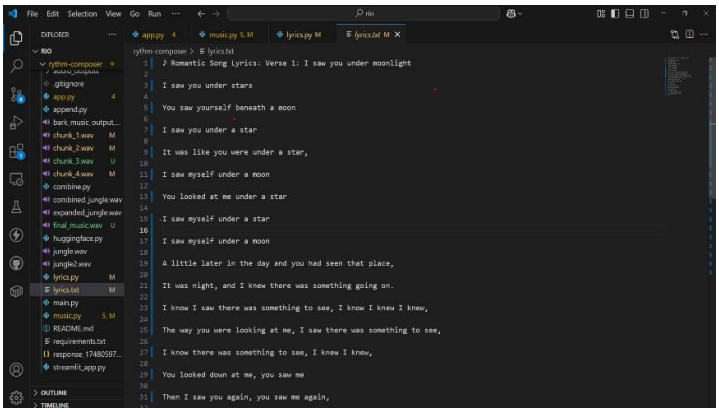


Figure 4: Output of Lyrics

After lyrics generation, music is produced in multiple chunks, each representing a segment of the final audio. These chunks are combined sequentially to form the complete music file, as shown in **Figure 5**, ensuring smooth transitions and coherent audio output.

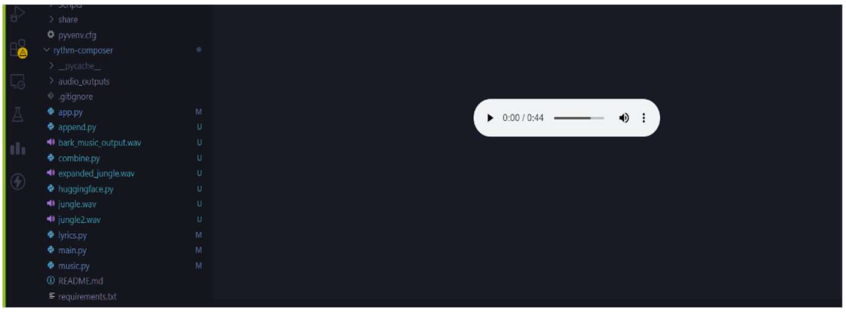


Figure 5: Generated Music Audio

5 Conclusion and Future Scope

The Rhythm Composer project presents the idea of applying artificial intelligence to make music by transforming a written source material into music in the form of a song, relying on the rhythm. It takes the GPT Neo to create lyrics together with the text-to-speech software to convert these lyrics to vocals. This assists the untrained music haters to develop complete music pieces. The design of the system which was composed of several components facilitated the construction and testing of different components individually. The first, user responses indicated that the songs sounded good and the lyrics fitted the rhythm enjoyably demonstrating that it can be used in learning and creative experiments. It also corrected quicker problem solving and upgrading since the developers could fine tune each module separately. The project identifies the issue of the modular AI design making the process less complex and enhancing the performance during integration of the models of various kinds.

The Rhythm Composer will be having more features in the future. These are incorporation of rules involving other types of music, real time sound creation and lyrics in numerous languages to enable more people use it. Additional features will enable the system to comprehend more emotions to sound more expressive and there will be enhanced music layering and harmony as well as coordination with other music production packages utilized by professionals. Such alterations will further enhance the strength and the utility of the tool by the creative as well as the professional users. Custom instrumentation and beats will also be added in order to provide users with greater scope of creative expression. These advancements will enable the tool to adjust to different musical styles and routines, thus being a useful addition to an artist, teacher or even a hobbyist.

References

[1] Transformer architecture introduced by Vaswani et al. (2017) Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[2] Music Transformer: Addressing long-term structure in music generation. Huang, C. Z. A., et al "Music transformer. arXiv preprint arXiv: 1809.04281." (2018).

[3] GPT family of models for text generation. Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[4] Jukebox: End-to-end music generation with vocals Dhariwal, Prafulla, et al. "Jukebox: A generative model for music. " arXiv preprint arXiv:2005.00341 (2020).

[5] Advances in text-to-audio models like AudioLM and suno-ai/bark Borsos, Zalán, et al. "Audiolm: a language modeling approach to audio generation."IEEE/ACM transactions on audio, speech, and language processing 31 (2023): 2523-2533.

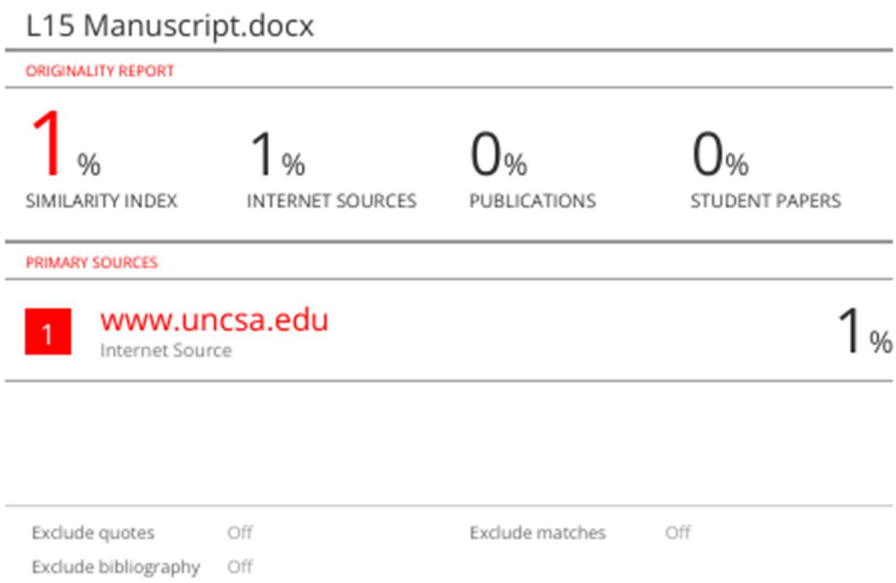
[6]
MusicLM: High-quality music generation from text descriptions. Agostinelli, Andrea, et al. "Musiclm: Generating music from text." arXiv preprint arXiv:2301.11325 (2023).

[7] MusicGen: Simple and Controllable Music Generation Copet, Jean, et al. "Simple and controllable music generation." arXiv preprint arXiv:2306.05284 (2023).

[8] MusicCraft: Structure-Aware Music Generation Ren, Yi, Jinzheng He, and Tao Qin. "MusicCraft: Structure-aware symbolic music generation." arXiv preprint arXiv:2309.10864 (2023).

[9] Hunyuan-Audio: Multi-level Language-Aligned Audio Generation Wu, Jiatong, et al. "Hunyuan-Audio: An audio foundation model for content creation with multiple levels of language-audio alignment." arXiv preprint arXiv:2310.02227 (2023).

Similarity Report



L15 Manuscript.docx

ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.uncsa.edu Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		