AI-powered Support System for Music Composition

Ngô Phi Hùng[1][your_orcid_id] and Phạm Quốc Vương[1][your_orcid_id]

University of Science, Ho Chi Minh City, Vietnam

Abstract. Nghiên cứu này tập trung vào việc sử dụng hai mô hình nhỏ gọn bằng kiến trúc hai lớp để thực hiện tác vụ sinh nhạc từ câu ngôn ngữ tiếng Anh hoặc tiếng Việt. Thay vì sử dụng kiến trúc một mô hình lớn truyền thống, chúng tôi thử nghiệm trên kiến trúc trên và điều chỉnh các tham số nhằm đánh giá khả năng sinh nhạc và so sánh với các mô hình hiện có. Kiến trúc kết hợp các mô hình ngôn ngữ lớn như GPT2, BERT cho từng tác vụ cụ thể gồm trích xuất đặc trưng và sinh nhạc Chúng tôi đã sử dụng BERT để làm mô hình trích xuất các đặc trưng về âm nhạc trong câu đầu vào và GPT-2 trong việc sinh nhạc, cùng với kỹ thuật giảm ma trận LoRA (Low-Rank Adaptation) để đơn giản hóa quá trình huấn luyện và tối ưu hóa hiệu suất mô hình. Chúng tôi cũng triển khai kỹ thuật kiểm tra tính đúng đắn và nội suy dựa trên nguyên tắc nhạc lý để đảm bảo bản nhạc hoàn chỉnh và sử dụng được. Đặc biệt, mô hình của chúng tôi cho thấy hiệu quả vượt trội hơn khoảng 12% so với MuseCoCo khi so sánh trên độ đo ASA[4], mặc dù chỉ sử dụng khoảng 200 triệu tham số khi thực nghiệm giữa hai mô hình. Tóm lại, nghiên cứu này chứng minh tiềm năng của các mô hình nhỏ gọn, kỹ thuật LoRA và GPT-2 trong việc sinh nhạc, mở ra nhiều hướng nghiên cứu và ứng dụng trong tương lai.

Keywords: text-to-midi · AI music generator.

1 Introduction

Trong những năm gần đây, nghiên cứu về mô hình hóa âm nhạc đã chứng kiến nhiều tiến bộ đáng kể với sự phát triển của các mô hình sinh nhạc nổi bật như MuseNet từ OpenAI, MusicGen từ Facebook, và MuseCoco từ Microsoft. Những công trình này đã cung cấp nền tảng quan trọng cho sự phát triển và cải tiến trong lĩnh vực, mở ra cơ hội mới cho các nghiên cứu tiếp theo.

Tuy nhiên, một thách thức phổ biến khi áp dụng các mô hình này là việc chuyển đổi các mô tả cảm tính của người dùng thành các đặc tính kỹ thuật cụ thể của bản nhạc. Người dùng thường mô tả bài hát dựa trên cảm xúc của họ, ví dụ: *"Một bài nhạc có tốc độ trung bình và cho cảm giác thư giản"*. Việc chuyển đổi những mô tả cảm tính này thành các thông số kỹ thuật cụ thể như tốc độ, giọng, và nhịp của bản nhạc gặp khó khăn do tính không rõ ràng và không nhất quán của văn bản mô tả. Điều này làm cho việc kiểm soát mô hình theo ý muốn và điều chỉnh nhạc theo ý muốn trở nên phức tạp và khó kiểm soát, đặc biệt đối với những người không có kiến thức kỹ thuật sâu về âm nhạc.

Để giải quyết vấn đề này, nghiên cứu hiện tại đề xuất phát triển các phương pháp sinh MIDI thay vì âm thanh. MIDI, với tính tiện dụng và khả năng điều chỉnh cao, có thể giúp tạo ra các bản nhạc mới một cách tự động và dễ dàng điều chỉnh theo phong cách âm nhạc cụ thể. Phương pháp này sẽ kết hợp mô tả kỹ thuật chi tiết với các đặc điểm cảm xúc của bài nhạc, nhằm tạo ra dữ liệu MIDI phù hợp với mục đích sáng tạo âm nhạc, mở rộng khả năng sáng tác và điều chỉnh âm nhạc hiệu quả hơn.

Ngoài ra, nghiên cứu cũng sẽ phát triển công cụ chuyển đổi cấu trúc dữ liệu giữa các mô hình cần thử nghiệm, nhằm tối ưu hóa quá trình thử nghiệm và so sánh hiệu suất của các mô hình khác nhau. Việc tiếp cận các mô hình được đào tạo trước như BERT và GPT-2 sẽ được thực hiện để thử nghiệm các tác vụ trích xuất đối tượng và sinh văn bản, giúp cải thiện khả năng hiểu và mô tả âm nhạc theo cách tự nhiên và chính xác hơn.

Bên cạnh đó, nghiên cứu sẽ tập trung vào các kỹ thuật như LoRA, Masked Language Modelling và Causal Language Modelling, kết hợp với các nghiên cứu hiện có, để phát triển một mô hình có khả năng hiểu văn bản mô tả âm nhạc theo cả khía cạnh kỹ thuật và cảm xúc. Điều này sẽ giúp tạo ra các bản nhạc dựa trên mô tả của người dùng một cách chính xác và sáng tạo hơn.

Mục tiêu của nghiên cứu là phát triển một hệ thống sinh MIDI đáp ứng nhu cầu của người làm nhạc, từ đó cung cấp một công cụ phù hợp cho ngành công nghiệp âm nhạc và công nghệ giải trí, đồng thời mở ra các tiềm năng ứng dụng rộng rãi trong lĩnh vực này.

2 Related Work

2.1 MuseCoco - Generating Symbolic Music from Text[4]

MuseCoco là hệ thống được đề xuất giúp sinh nhạc ở dạng MIDI từ văn bản mô tả các đặc điểm mang tính kỹ thuật của bản nhạc, ví dụ: *"Bản nhạc có nhịp 4/4, viết ở giọng la thứ, có các nhạc cụ piano, guitar và sáo kết hợp với nhau"*. Kiến trúc của MuseCoco gồm hai mô hình: một mô hình để hiểu văn bản mô tả âm nhạc và một mô hình để sinh nhạc.

2.2 MusicLM: Generating Music From Text[1]

MusicLM là một mô hình được phát triển bởi Google Research, có khả năng tạo ra âm nhạc chất lượng cao dựa trên mô tả bằng văn bản, ví dụ: *"Bản nhạc Pop có giọng nữ nhẹ nhàng hát trên phần đệm lead synth*[1] *đầy đặn, giai điệu piano êm dịu, tiếng kèn đồng ngân dài, và tiếng trống mạnh mẽ, cùng cảm giác buồn, luyến nhớ, làm người nghe liên tưởng đến những bản nhạc thường phát trên radio."*. Ngoài ra, MusicLM có thể xử lý đồng thời cả văn bản và giai điệu, nghĩa là nó nhận đầu vào là các giai điệu được huýt sáo hoặc ngân nga và văn bản mô tả bổ sung để cho ra giai điệu mới. Đầu ra của mô hình này có dạng audio.

[1] Còn gọi là lead synthesizer - một loại nhạc cụ điện tử dùng để đệm các tuyến giai điệu chính của bản nhạc.

2.3 MUGEN: A Playground for Video-Audio-Text Multimodal Understanding and GENeration[2]

MuGen là mô hình được nghiên cứu để hiểu và sinh âm thanh cho video game dựa trên đầu vào là video của một cảnh game và đoạn văn bản mô tả, ví dụ: *"Nhân vật chạy đến bên phải để thu thập đồng xu. Sau đó, nhân vật bị nảy lên và rơi trúng quái vật ốc sên khiến nó bị tiêu diệt."*. Đầu ra của mô hình này có dạng audio.

Nhận xét:

So với những nghiên cứu trên, mô hình của chúng tôi đã áp dụng những ưu điểm của các mô hình có sẵn và hạn chế những khuyết điểm của chúng như sau:

1. Sử dụng MIDI là dạng dữ liệu cho kết quả đầu ra: Định dạng này có thể chỉnh sửa được và phù hợp với nhiều mục đích khác nhau như thay đổi nhạc cụ, thay đổi giai điệu và hoà âm, thay đổi tiết tấu của đoạn nhạc, v.v. Ngoài ra kích thước dữ liệu nhỏ gọn của MIDI giúp giảm đáng kể các chi phí trong quá trình huấn luyện so với tập tin audio.
2. Tận dụng kiến trúc hai mô hình huấn luyện riêng biệt cho từng tác vụ của MuseCoco[4]: Việc này giúp mỗi mô hình trở nên gọn nhẹ hơn mà vẫn đảm bảo hiệu năng, giúp tối ưu hóa tài nguyên và tăng hiệu quả xử lý. Tuy nhiên, chúng tôi áp dụng thêm kỹ thuật LoRA để tối ưu hoá tài nguyên và chi phí huấn luyện so với MuseCoco.
3. Khả năng sinh ra nội dung dài: Mô hình của chúng tôi có thể tạo ra nội dung dài hơn so với các nghiên cứu có trước, giúp đáp ứng tốt hơn nhu cầu của người dùng.
4. Ngoài ra, chúng tôi bổ sung thêm lượng lớn dữ liệu để phục vụ cho việc huấn luyện mô hình.

3 Proposed Method

3.1 Overview

Bài toán biến đổi ngôn ngữ tự nhiên thành âm nhạc đòi hỏi một mô hình có khả năng tạo ra bản nhạc từ văn bản mô tả đầu vào cụ thể. Âm nhạc sinh ra phải phù hợp với nội dung câu văn đầu vào, lẫn mang tính thẩm mỹ, sáng tạo và hấp dẫn về mặt âm nhạc. Để đạt được điều đó, chúng tôi đã thực hiện lựa chọn kiến trúc mô hình phù hợp và chuẩn bị nguồn dữ liệu chất lượng cao.

Về mô hình Chúng tôi sử dụng các kỹ thuật huấn luyện Masked Language Modelling, Casual Language Modelling và LoRA làm nền tảng. Đồng thời tinh chỉnh các siêu tham số hợp lý để cho ra kết quả mong muốn. Kiến trúc của hệ thống bao gồm hai mô hình tương ứng với hai giai đoạn:

Giai đoạn 1: Trích xuất các thuộc tính âm nhạc

Với mô hình tương ứng là **text2att**, giai đoạn 1 tập trung vào tác vụ xác định các đặc trưng âm nhạc dựa trên văn bản mô tả âm nhạc đầu vào, bằng việc xác định và phân loại nhãn. Các nhãn này đại diện cho các thuộc tính định tính và định lượng của đoạn nhạc cần tạo ra. Ví dụ: Trong đoạn nhạc có sử dụng nhạc cụ nào (piano, guitar, etc); Tốc độ của đoạn nhạc như thế nào (nhanh, vừa, chậm, etc); Tốc độ của đoạn nhạc là bao nhiêu (87BPM, 100BPM, 128BPM, etc); Cao độ bản đoạn phân bố trong bao nhiêu quãng tám; Đoạn nhạc sử dụng nhịp gì ($\frac{3}{4}$, $\frac{4}{4}$, $\frac{6}{8}$, etc); etc.

Giai đoạn 2: Sinh nhạc từ các nhãn đã trích xuất

Với mô hình tương ứng là **att2midi**, giai đoạn 2 tập trung vào tác vụ sinh nhạc. Đầu vào của giai đoạn này là các nhãn được xác định từ văn bản mô tả âm nhạc ở giai đoạn 1. Một câu prompt tương ứng với các nhãn sẽ được tạo ra để giúp mô hình xác định được các đặc tính của đoạn nhạc cần tạo.

Kiến trúc hai lớp trên giúp tạo ra tính độc lập trong quá trình huấn luyện hai mô hình. Điều đó không chỉ giúp đạt được các lợi ích về giảm thiểu chi phí và tài nguyên huấn luyện, mà còn giúp đơn giản hoá quá trình chuẩn bị dữ liệu do nỗi mô hình đều có dữ liệu mục tiêu không phụ thuộc quá cao vào nhau. Tuy mỗi giai đoạn có một mô hình độc lập để giải quyết tác vụ tương ứng nhưng chúng vẫn kết hợp tốt do đầu ra của mô hình ở giai đoạn 1 là đầu vào của mô hình ở giai đoạn 2.

Về dữ liệu Tương ứng với hai mô hình ở hai giai đoạn đã nêu, chúng tôi sử dụng hai định dạng dữ liệu chính sau để phục vụ cho quá trình huấn luyện:

Định dạng dữ liệu cho mô hình text2att: Các template mô tả âm nhạc ở dạng ngôn ngữ tự nhiên (gồm các template tiếng Anh và các template tiếng Việt tương ứng). Bên cạnh ngôn ngữ tự nhiên, mỗi template chứa placeholder cho nhãn mục tiêu (nếu có). Một template tiếng Anh mẫu: *"The song has a fast tempo and a [TIME_SIGNATURE] time signature. It is bright at its start but then turns dark. The tune is played with [INSTRUMENTS]."*

Định dạng dữ liệu cho mô hình att2midi: Các cặp "source - target", nói cách khác là "command - music". Trong đó, `music` là dữ liệu âm nhạc dạng MIDI được token hoá thành âm nhạc dạng văn bản dựa trên REMI[3] của MuseCoco[4]. Ví dụ: "`s-19 o-0 t-44 i-0 p-62 d-6 v-20 p-50 d-11 v-20`". Command là thông tin metadata, của đoạn nhạc MIDI, được đưa về dạng tương tự ví dụ sau (xem ý nghĩa các trường dữ liệu ở bảng 1):

$$
\begin{aligned}
I1s2 &: (11, 4) \\
I4 &: (11, \text{False}) \\
R3 &: 1 \\
B1s1 &: (16, 3) \\
TS1s1 &: (4, 4) \\
K1 &: \text{major} \\
T1s1 &: (114.03503592196344, 1) \\
P4 &: 3 \\
TM1 &: (33.45463058047076, 2)
\end{aligned}
$$

Table 1. Metadata tương ứng với các trường dữ liệu trong command

Ký hiệu	**Metadata**
I1s2	Instrument
I4	Main Instrument
R3	Rhythm Intensity
B1s1	Bar
TS1s1	Time Signature
K1	Key
T1s1	Tempo
P4	Pitch Range
TM1	Time

3.2 Data Preparation

Tương ứng với hai định dạng dữ liệu cho hai mô hình đã nêu ở mục 3.1, phần này chúng tôi mô tả quá trình thu thập và tiền xử lý dữ liệu ngôn ngữ tự nhiên và dữ liệu âm nhạc.

Dữ liệu ngôn ngữ tự nhiên cho mô hình text2att Mục tiêu đầu ra của công đoạn chuẩn bị dữ liệu ngôn ngữ tự nhiên là các template mô tả âm nhạc bằng tiếng Anh và tiếng Việt. Để đáp ứng nhu cầu dữ liệu tiếng Anh, chúng tôi tận dụng lại 4815 template do nhóm tác giả bài báo MuseCoco tạo ra bằng việc sử dụng ChatGPT trong quá trình nghiên cứu. Ngoài ra, chúng tôi cũng sử dụng prompt engineering với ChatGPT để bổ sung thêm dữ liệu, nâng tổng số mẫu dữ liệu tiếng Anh thành 14900 mẫu, sau đó sử dụng kỹ thuật dịch văn bản

do chúng tôi đề xuất để có thêm 14900 mẫu dữ liệu tiếng Việt tương ứng. Danh sách các nhãn trong câu template và giá trị tương ứng được mô tả ở bảng 2.

Table 2. Tên và giá trị tương ứng của các nhãn trong mỗi câu template từ bài báo MuseCoco[4] (đã được rút gọn dựa trên các nhãn chúng tôi sử dụng)

Tên nhãn	**Giá trị**
Instrument	28 instruments: piano, keyboard, percussion, organ, guitar, bass, violin, viola, cello, harp, strings, voice, trumpet, trombone, tuba, horn, brass, sax, oboe, bassoon, clarinet, piccolo, flute, pipe, synthesizer, ethnic instrument, sound effect, drum. Mỗi nhạc cụ: 0: Được chơi, 1: Không được chơi, 2: NA
Pitch	Range: 0-11: octaves, 12: NA.
Rhythm Danceability	0: danceable, 1: not danceable, 2: NA.
Bar	0: 1-4 bars, 1: 5-8 bars, 2: 9-12 bars, 3: 13-16 bars, 4: NA.
Time Signature	0: 4/4, 1: 2/4, 2: 3/4, 3: 1/4, 4: 6/8, 5: 3/8, 6: các nhịp khác, 7: NA.
Key	0: major, 1: minor, 2: NA.
Tempo	0: chậm (<=76 BPM), 1: trung bình (76-120 BPM), 2: nhanh (>=120 BPM), 3: NA.
Time	0: 0-15s, 1: 15-30s, 2: 30-45s, 3: 45-60s, 4: >60s, 5: NA.

Dữ liệu âm nhạc cho mô hình att2midi Mục tiêu đầu ra của công đoạn chuẩn bị dữ liệu âm nhạc là cặp "command - music" như đã nêu ở mục 3.1. Ngoài tái sử dụng 300 mẫu được công khai của MuseCoco[4], chúng tôi thực hiện thu thập dữ liệu từ Hooktheory[2] - trang web chuyên cung cấp sách điện tử, bài viết, thống kê, phần mềm giáo dục về lý thuyết âm nhạc, cũng như thông tin ký âm và hoà âm của hơn 40000 bản nhạc trên thế giới. Sau qua trình thu thập, chúng tôi thực hiện các bước xử lý từ dữ liệu thu thập được thành 29000 đoạn nhạc ở dạng tập tin MIDI. Các tập tin MIDI này được biến đổi về dạng "command - music" theo ba bước sau:

1. Đầu tiên, sử dụng thư viện midi_data_extractor của MuseCoco[4] để trích xuất thông tin metadata (nhịp, tốc độ bản nhạc, nhạc cụ, v.v) từ một tập tin MIDI. Metadata này sau đó được ánh xạ với bộ từ điển âm nhạc cho trước để tìm ra các thông tin như thời lượng bài nhạc, tốc độ, nhạc cụ được chơi, và các chi tiết khác (xem các trường metadata ở bảng 1).

[2] Liên kết đến trang web: https://www.hooktheory.com, truy cập lần cuối 23/07/2024.

2. Phần command được tạo ra bằng cách ghép những metadata này thành các câu prompt phù hợp với bài nhạc. Đây là bản command để điều khiển sinh nhạc.
3. Sau đó, tập tin MIDI được chuyển thành dạng văn bản theo phương pháp token hoá dựa trên REMI[3] của MuseCoco[4]. Đây sẽ là phần music trong định dạng dữ liệu mục tiêu.

3.3 Mô Hình text2att

Mô hình 1 Chúng tôi sử dụng mô hình BERT, một mô hình thuộc loại encoder trong việc trích xuất đặc trưng văn bản, để dự đoán các thuộc tính âm nhạc từ câu văn. Chúng tôi phát triển một biến thể **MusicBERT**[6], được tinh chỉnh để phù hợp với nhiệm vụ phân loại các thuộc tính âm nhạc đa dạng. Các thuộc tính này bao gồm cả định tính (như sự xuất hiện của các nhạc cụ) và định lượng (như thời lượng và tốc độ của bản nhạc).

Mô hình MusicBERT[6] được phát triển từ kiến trúc BERT gốc, với các tinh chỉnh để phân loại các thuộc tính âm nhạc. Mô hình thêm các token [CLS_i] chính là các thông tin của các nhãn đã để cập ở bảng trên. Sau khi đi qua BERT để trích xuất ra các đặc trưng, đầu ra là các logits, sau đó sẽ đi qua một lớp Softmax để phân loại nhãn.

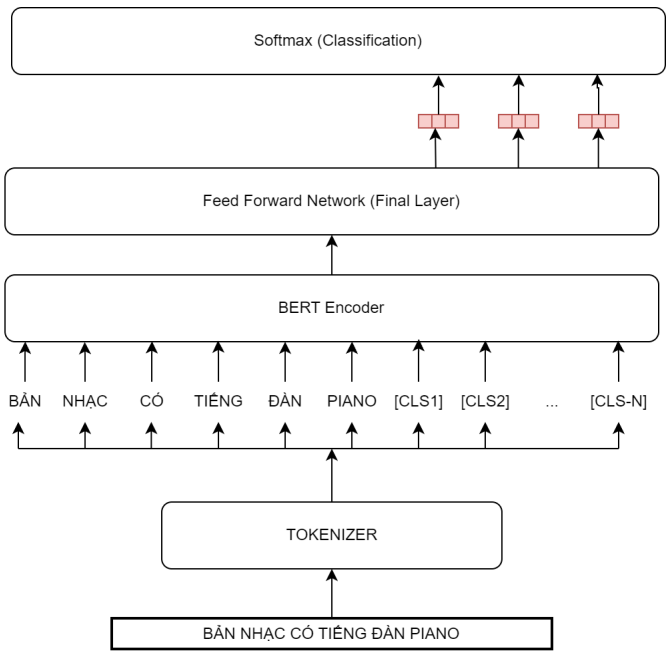

Fig. 1. Kiến trúc mô hình text2att sử dụng BERT

3.4 Mô hình sinh nhạc

Khi bắt đầu nghiên cứu về mô hình ngôn ngữ, một trong những thách thức lớn nhất là lựa chọn kiến trúc và mô hình phù hợp với nguồn tài nguyên sẵn có mà vẫn đảm bảo kết quả tốt. Bài toán đặt ra là cân bằng giữa hiệu suất của mô hình và tài nguyên tính toán. Nếu mô hình không đủ mạnh, kết quả sẽ không đáp ứng được mục tiêu đề ra. Ngược lại, nếu sử dụng mô hình với nhiều tham số, lượng tài nguyên cần thiết sẽ rất lớn. Bài luận này sẽ thảo luận về các yếu tố cần xem xét khi lựa chọn mô hình ngôn ngữ, cũng như các giải pháp tiềm năng để tối ưu hóa tài nguyên và hiệu suất. Trong ngữ cảnh này, sự ra đời của Low-Rank Adaptation (LoRA) đã mang lại một giải pháp hiệu quả để giải quyết vấn đề này.

Sử dụng kiến trúc GPT-2 được tối ưu bằng cách áp dụng phương pháp LoRA. Mô hình này được thiết kế để tăng khả năng mở rộng và linh hoạt của GPT-2 mà không làm tăng đáng kể số lượng tham số cần huấn luyện, giúp giảm thiểu yêu cầu về tài nguyên tính toán mà vẫn giữ được hiệu suất cao.

Trong bài toán sinh nhạc, việc chuẩn bị dữ liệu đầu vào cho mô hình là một bước quan trọng để đảm bảo hiệu suất và độ chính xác của quá trình huấn luyện. Tokenizer là công cụ phân tích và chuyển đổi dữ liệu đầu vào thành các đơn vị nhỏ hơn (token) để mô hình có thể xử lý hiệu quả.

Bộ dữ liệu âm nhạc có một bộ từ vựng và kiểu dữ liệu đặc trưng riêng, bao gồm các thuộc tính như nhạc cụ, cao độ, nhịp điệu, cường độ nhịp, ô nhịp, nhịp, khóa, tốc độ, thời gian. Mỗi thuộc tính này cần được mã hóa chính xác để mô hình có thể hiểu và học từ dữ liệu. Mô hình được huấn luyện theo kỹ thuật CLM (Casual Language Modelling) dựa trên tập dữ liệu command-music.

Kỹ thuật LoRA được cấu hình để cải thiện hiệu suất của mô hình. Các tham số như r, $lora\_alpha$, $lora\_dropout$, và các module đích được thiết lập để giảm số lượng tham số cần huấn luyện và tăng tốc quá trình huấn luyện.

Fig. 2. Mô hình sinh nhạc dạng command-music

3.5 Hậu xử lý

Một mô hình sinh dù được huấn luyện tốt đến đâu, vẫn luôn tồn tại rủi ro dữ liệu sinh ra không thể mã hóa thành thông tin mong muốn. Do đó, chúng tôi đề xuất một thuật toán kiểm tra và sửa lỗi để đảm bảo kết quả của mô hình luôn có thể chuyển thành dạng dữ liệu cuối cùng là MIDI.

Bằng việc xem tập tin MIDI là một kiểu thể hiện của dữ liệu time series, chúng tôi kiểm tra từng vị trí tương ứng với các thời điểm có sự kiện MIDI như

tempo thay đổi, time signature thay đổi, hoặc có nốt mới xuất hiện. Chúng tôi xem xét từng sự kiện đó có tuân theo những quy luật cho trước hay không. Các quy luật này được ánh xạ từ bộ quy tắc dựa trên REMI[3] của MuseCoco[4] (xem đặc tả dữ liệu âm nhạc dạng văn bản của MuseCoco ở bảng 3).

Table 3. Đặc tả dữ liệu âm nhạc dạng văn bản của MuseCoco

Thuộc tính	Tên đầy đủ	Mô tả	Thuộc tính liền sau
s	Time Signature	Nhịp của bản nhạc	b, o.
o	Position	Thời điểm xuất hiện của sự kiện	t.
t	Tempo	Tốc độ bản nhạc	i.
i	Instrument	Nhạc cụ	p.
p	Pitch	Cao độ nốt nhạc	d.
d	Duration	Độ dài nốt nhạc	v.
v	Velocity	Tốc độ nhấn/Lực nhấn phím đàn	i, b, p, o.
b	Bar	Vạch nhịp	s.

Sau khi kiểm tra toàn bộ các sự kiện, nếu có sự kiện lỗi, thông thường chúng sẽ bị bỏ qua, nhưng điều này có thể làm mất tính mạch lạc của bài nhạc. Nhằm khắc phục điều đó, chúng tôi thực hiện hai bước chính (xem ví dụ minh hoạ ở hình 3):

1. **Khôi phục cấu trúc:** Với các sự kiện bị phá vỡ cấu trúc, chúng tôi thay thế bằng cấu trúc của sự kiện gần nhất có cấu trúc tương-tự-và-đúng. Nếu toàn bộ dữ liệu không có sự kiện đúng, một cấu làm chuẩn tương ứng với sự kiện do chúng tôi đề ra từ đặc tả dữ liệu sẽ được áp dụng.
2. **Khôi phục giá trị:** Giá trị của các thành phần cấu trúc trong sự kiện lỗi sẽ được điền vào cấu trúc mới đã nêu để duy trì thông tin chính xác nhất có thể. Nếu có thành phần cấu trúc còn thiếu giá trị, chúng tôi sẽ chọn giá trị điền vào theo thứ tự ưu tiên sau: giá trị của thành phần tương ứng gần nhất trong toàn bộ đoạn nhạc; giá trị mặc định cho trước[3].

[3] Với pitch, việc chọn giá trị thay thế có thể thay bằng cách: Xác định key signature của tập hợp các nốt không bị lỗi, sau đó, điền bằng giá trị pitch trung bình giữa hai nốt trước và sau nốt lỗi (làm tròn đến giá trị pitch gần nhất trong key).

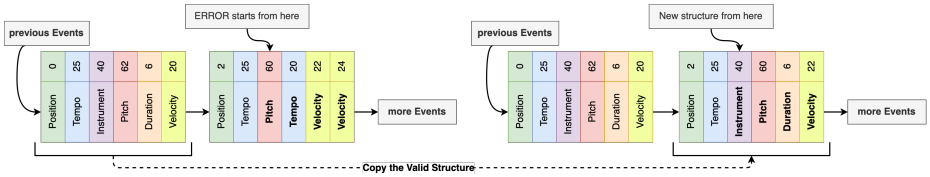

Fig. 3. Ví dụ minh hoạ quá trình hậu xử

Nhờ vào thuật toán này, chúng tôi có thể giảm thiểu các sai sót và đảm bảo đầu ra dữ liệu cuối cùng luôn hợp lệ và mạch lạc nhất có thể.

4 Experiments

4.1 Experiment Setup

System Configuration text2att Để huấn luyện giai đoạn text2att. Các siêu tham số được sử dụng trong quá trình huấn luyện mô hình BERT bao gồm: số lượng epoch là 100, xác định số lần mô hình sẽ được huấn luyện trên toàn bộ dữ liệu. Kích thước batch là 32, tức là kích thước của mỗi batch dữ liệu được đưa vào mô hình trong mỗi bước huấn luyện. Bộ tối ưu hóa sử dụng là AdamW, dùng để cập nhật các tham số của mô hình trong quá trình huấn luyện. Tốc độ học (learning rate) là 2e-5, xác định mức độ điều chỉnh các trọng số của mô hình sau mỗi bước huấn luyện. Chiều dài tối đa của chuỗi đầu vào (max sequence length) là 128, các chuỗi dài hơn sẽ bị cắt ngắn. Số bước warmup là 500, tức là số bước đầu tiên trong đó tốc độ học tăng dần đến giá trị tối đa đã định. Tỷ lệ dropout là 0.1, sử dụng trong quá trình huấn luyện để tránh overfitting. Số bước gradient accumulation là 2, tức là số bước gradient accumulation trước khi cập nhật trọng số mô hình. Hệ số weight decay là 0.01, giúp điều chỉnh tỷ lệ suy giảm trọng số, giúp tránh overfitting.

System Configuration att2midi Trong quá trình huấn luyện mô hình GPT-2 sử dụng kỹ thuật LoRA (Low-Rank Adaptation), chúng tôi đã thực hiện các bước chi tiết từ chuẩn bị dữ liệu, token hóa, cấu hình mô hình, áp dụng kỹ thuật LoRA, đến huấn luyện và đánh giá kết quả. Dữ liệu được token hóa với các tham số như độ dài tối đa và padding, sau đó được chia theo tỷ lệ 80:10:10 cho các tập train, valid và test. Mô hình GPT-2 được cấu hình với 20 lớp transformer, 16 head trong multi-head attention, kích thước embedding 1024, kích thước từ vựng 1253, và số lượng vị trí tối đa là 2048. Kỹ thuật LoRA được áp dụng nhằm giảm số lượng tham số cần huấn luyện, giúp tăng tốc quá trình huấn luyện. Các tham số cụ thể cho LoRA bao gồm $r = 16$, $\alpha = 12$, $dropout = 0.1$, và các module đích như c_proj, c_attn, wte, lm_head . Quá trình huấn luyện và đánh giá mô hình được thực hiện với các tham số như số epoch là 10, batch size là 8, optimizer AdamW, learning rate 2×10^{-4}, max sequence length 2048, warmup steps 2000, dropout rate 0.1, gradient accumulation steps 2, và weight decay 0.01. Mô hình

được huấn luyện sử dụng lớp Trainer của transformers với tổng số tham số là **203 triệu**. Số lượng tham số được huấn luyện là gần **3.5 triệu**.

Evaluation Để đánh giá mô hình trên, chúng tôi xây dựng một bộ kiểm tra tiêu chuẩn bao gồm 1.000 mẫu ở hai bộ data cho hai mô hình text2att và att2music. Ở mô hình text2att chúng tôi kiểm tra bằng cách so khớp nhãn có sẵn với nhãn mà mô hình dự đoán tương tự các mô hình phân loại. Với mô hình att2midi, chúng tối tôi phải thông qua một bước trích xuất metadata của bản nhạc sau khi tạo ra để so khớp mới các nhãn đầu vào. Trên đây là cách đánh giá chi tiết riêng rẻ từng mô hình.

Chúng tôi đã đánh giá cụ thể từng loại nhãn như các mô hình phân loại bằng cách lấy

$$accuracy = \frac{\text{số dự đoán đúng}}{\text{tổng số dự đoán}}$$

Để đánh giá các mô hình một cách khách quan, chúng tôi đã tham khảo tiêu chí có tên là Average Sample-wise Accuracy - ASA, được đề xuất trong bài báo MuseCoco[4], được tính bằng cách xác định tỷ lệ thuộc tính được dự đoán chính xác trong mỗi mẫu, sau đó tính trung bình độ chính xác dự đoán trên toàn bộ bộ kiểm tra ở bài báo MuseCoco.

Ngoài ra còn có một số tiêu chí đánh giá như:

1. Average pitch count[5]: Đây là độ đo nhằm hỗ trợ xác định sự phân phối cao độ (pitch) của 128 cao độ mà tập tin MIDI hỗ trợ. Công thức:

$$average\_pitch\_count = \frac{\text{số lần xuất hiện của cao độ đang xét}}{\text{tổng số nốt trong tập tin MIDI}}$$

2. Average pitch class count[5]: Đây là độ đo nhằm hỗ trợ xác định sự phân phối cao độ (pitch) của 12 nốt nhạc cơ bản. Công thức:

$$average\_pitch\_class\_count = \frac{\text{số lần xuất hiện}}{\text{tổng số nốt trong tập tin MIDI}},$$

   với số lần xuất hiện được tính bằng số các nốt có cao độ đồng dư với cao độ đang xét theo modulo 12.
3. Subjective evaluation: Đánh giá dựa trên cảm nhận người nghe.

4.2 Main result

Kết quả được tổng hợp sau quá trình huấn luyện và dự đoán với bộ checkpoint tốt nhất và thực hiện đánh giá trên tập kiểm tra gồm 1000 mẫu.

Với các thuộc tính âm nhạc liên quan đến các loại nhạc cụ (instrument), khi đánh giá, chúng tôi chỉ tập trung vào các loại nhạc cụ phổ biến trong dữ liệu huấn luyện. Kết quả được đo lường và cho thấy độ chính xác rất cao (xem bảng 4).

Cụ thể, các loại nhạc cụ như accordion, brass, celesta, choir, guitar, harmonica, organ, piano, synth, viola, violin, và voice đều đạt độ chính xác cao, dao động từ 0.90 đến 0.99 trong cả tiếng Anh (ENG) và tiếng Việt (VIE). Điều này cho thấy mô hình trích xuất đặc trưng của chúng tôi hoạt động tốt và nhất quán với các loại nhạc cụ này, bất kể ngôn ngữ đánh giá.

Table 4. Instrument Accuracy

Instrument	**Accuracy (ENG)**	**Accuracy (VIE)**
accordion	0.94	0.93
brass	0.98	0.96
celesta	0.91	0.92
choir	0.95	0.97
guitar	0.99	0.93
harmonica	0.97	0.94
organ	0.90	0.91
piano	0.96	0.95
synth	0.92	0.94
viola	0.91	0.90
violin	0.93	0.92
voice	0.95	0.96

Với các thuộc tính âm nhạc khác, kết quả được đo lường cho thấy độ chính xác rất cao, các thuộc tính như **Time Signature** và **Pitch Range** đạt độ chính xác cao nhất, lần lượt là 0.94 (tiếng Anh - ENG) và 0.94 (tiếng Việt - VIE). Điều này cho thấy mô hình có khả năng phân loại tốt các yếu tố liên quan đến nhịp điệu và phạm vi âm thanh. Các thuộc tính như **Rhythm Danceability** và **Tempo** có độ chính xác thấp hơn, dao động từ 0.85 đến 0.93, nhưng vẫn cho thấy mức độ chính xác cao và đáng tin cậy (xem bảng 5).

Table 5. Categories Accuracy

Category	**Accuracy (ENG)**	**Accuracy (VIE)**
Rhythm Danceability	0.85	0.87
Bar	0.91	0.89
Time Signature	0.94	0.92
Key	0.88	0.90
Tempo	0.93	0.85
Pitch Range	0.90	0.94

So sánh với mô hình MuseCoCo ở cùng phần cấu hình, dữ liệu và lượng dữ liệu huấn luyện chúng tôi thu được một số đnáh giá như sau: (xem bảng 6).

Nhận xét mô hình Độ chính xác của các thuộc tính âm nhạc giữa mô hình chúng tôi đề xuất và MuseCoCo thể hiện ở bảng 6 cho thấy mô hình LoRA GPT2 vượt trội hơn về hầu hết các độ đo. Đặc biệt, trong các độ đo ASA và Rhythm Danceability, LoRA GPT2 có độ chính xác rất cao, lần lượt là 0.64 và 0.96. Tuy nhiên, đối với độ đo Time Signature, MuseCoCo có kết quả cao hơn với giá trị 0.43 so với 0.37. Nhìn chung, mô hình chúng tôi thể hiện hiệu quả tốt hơn trong việc dự đoán các thuộc tính âm nhạc phổ biến.

Table 6. So sánh kết quả giữa hai mô hình.

Độ đo	**LoRA GPT2**	**Kiến trúc MuseCoCo**
ASA	**0.64**	0.57
Instrument	0.67	0.6
Pitch Range	0.68	0.52
Rhythm Danceability	0.96	0.89
Bar	0.51	0.42
Time Signature	0.37	0.43
Key	0.57	0.51
Tempo	0.69	0.61

Đánh giá đầu ra dựa trên các thông số về cao độ nốt nhạc Xem các hình 4, 5, 6, và 7.

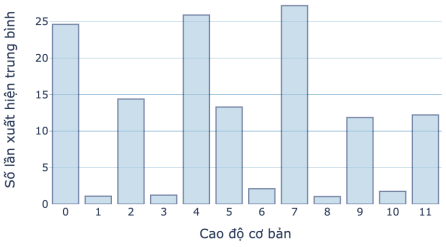

Fig. 4. Số lần xuất hiện trung bình của các giá trị cao độ cơ bản (LoRA GPT2)

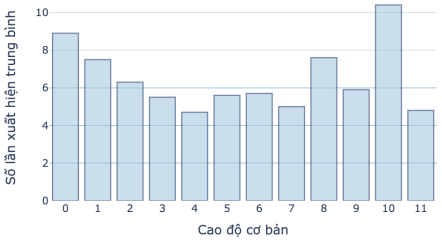

Fig. 5. Số lần xuất hiện trung bình của các giá trị cao độ cơ bản (Kiến trúc MuseCoco)

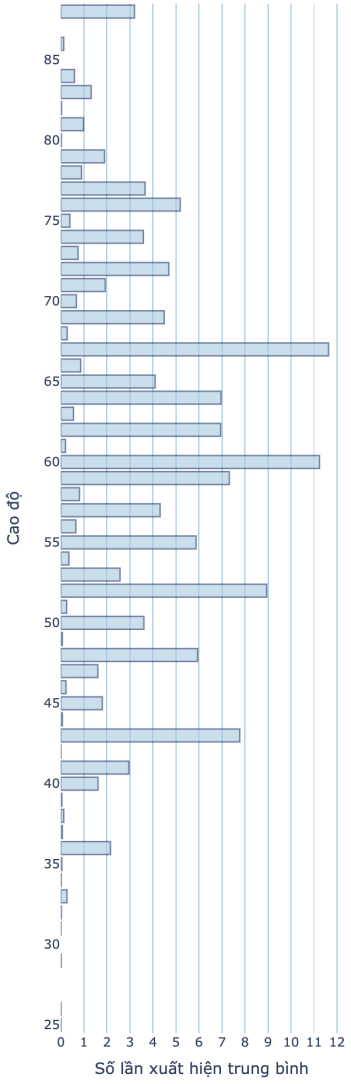

Fig. 6. Số lần xuất hiện trung bình của các giá trị cao độ (LoRA GPT2)

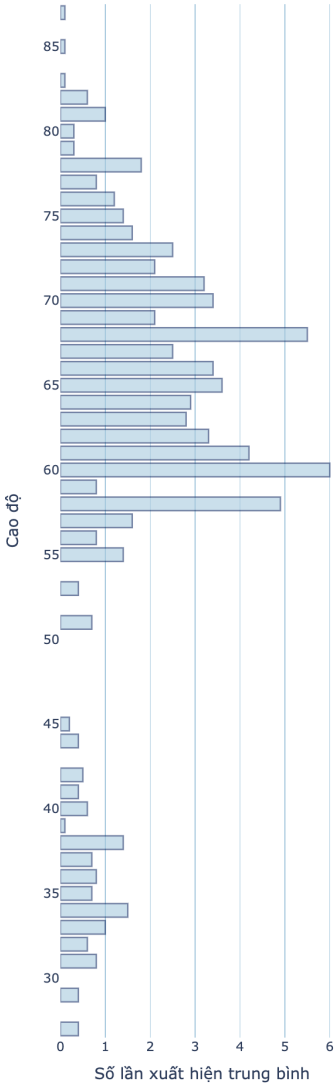

Fig. 7. Số lần xuất hiện trung bình của các giá trị cao độ (Kiến trúc MuseCoco)

Đánh giá từ người có kiến thức về âm nhạc Âm nhạc được tạo ra khá tự nhiên. Các nốt nhạc khi nghe không bị sai scale. Công cụ này tốt cho việc cung cấp cảm hứng ban đầu trong việc sáng tạo âm nhạc. Điều đó giúp tăng hiệu quả sáng tác, tiết kiệm được thời gian làm việc để giành cho các công đoạn phức tạp sau đó như viết lời, arrangement, mixing, mastering, v.v.

5 Conclusion

Chúng tôi đã tiến hành nghiên cứu và thực nghiệm phương pháp sử dụng hai mô hình nhỏ gọn để xử lý tác vụ sinh nhạc từ câu ngôn ngữ tiếng Anh hoặc tiếng Việt. Thay vì sử dụng một mô hình lớn như truyền thống, chúng tôi thử nghiệm trên kiến trúc mô hình hai lớp và điều chỉnh nhiều tham số khác nhau để đánh giá khả năng sinh nhạc, qua đó đạt được một số kết quả cụ thể.

Kết quả nghiên cứu cho thấy việc sử dụng các mô hình decoder như GPT-2 rất phù hợp với quá trình sinh nhạc, trong khi mô hình encoder như BERT phù hợp với việc phân loại và trích xuất các đặc trưng. Chúng tôi cũng đã áp dụng kỹ thuật LoRA trong quá trình huấn luyện nhằm tối ưu hóa nguồn tài nguyên. Ngoài ra, chúng tôi triển khai kỹ thuật kiểm tra tính đúng đắn và nội suy dựa trên nguyên tắc nhạc lý để đảm bảo bản nhạc hoàn chỉnh và có thể ứng dụng được trong sản phẩm.

Mặc dù các phương pháp trên đều cho kết quả tương đối tốt, vẫn cần nghiên cứu thêm để điều chỉnh các bộ tham số trên các tập dữ liệu phong phú và đa dạng hơn. Mục tiêu là đạt được mô hình hỗ trợ nhiều thể loại và màu sắc âm nhạc khác nhau. Các thực nghiệm trên cũng có thể mở rộng ra bằng việc sử dụng nhiều tham số hơn trong các mô hình.

Tóm lại, nghiên cứu này đã chứng minh tiềm năng của các mô hình nhỏ gọn và kỹ thuật LoRA trong việc sinh nhạc, mở ra nhiều hướng nghiên cứu và ứng dụng trong tương lai.

6 Acknowledgment

Chúng tôi xin bày tỏ sự trân trọng và lòng biết ơn đến thầy Trần Duy Hoàng - University of Science, Ho Chi Minh, Vietnam, với sự hỗ trợ trong quá trình dịch thuật, review và đề xuất chỉnh sửa để bài báo được hoàn thiện nhất có thể.

References

1. Agostinelli, A., Denk, T.I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., Sharifi, M., Zeghidour, N., Frank, C.: Musiclm: Generating music from text (2023)
2. Hayes, T., Zhang, S., Yin, X., Pang, G., Sheng, S., Yang, H., Ge, S., Hu, Q., Parikh, D.: Mugen: A playground for video-audio-text multimodal understanding and generation (2022)
3. Huang, Y.S., Yang, Y.H.: Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions (2020)
4. Lu, P., Xu, X., Kang, C., Yu, B., Xing, C., Tan, X., Bian, J.: Musecoco: Generating symbolic music from text (2023)
5. Warnerfjord, M.: Evaluating chatgpt's ability to compose music using the midi file format (2023)
6. Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T., Liu, T.Y.: Musicbert: Symbolic music understanding with large-scale pre-training (2021), https://arxiv.org/abs/2106.05630