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. Hiện nay, bài toán sinh nhạc từ văn bản mô tả có hai hướng giải quyết chính: text-to-audio và text-to-symbolic-music. Một trong những key difference quan trọng nhất của hai hướng trên là sự đánh đổi giữa khả năng cho phép chỉnh sửa một cách chi tiết và độ giống âm nhạc thực tế của kết quả đầu ra. Nói cách khác, khi đã được tạo ra, trái ngược với symbolic music, music in audio formats giống với âm nhạc thực tế hơn, nhưng không cho phép chỉnh sửa các nốt nhạc bên trong. Với mục tiêu tạo ra một hệ thống AI sinh nhạc có đầu ra có thể chỉnh sửa một cách chi tiết, nghiên cứu này được thực hiện theo hướng text-to-midi - một dạng của text-to-symbolic-music. Thay vì sử dụng một mô hình lớn theo cách truyền thống, chúng tôi sử dụng kiến trúc hai lớp với hai mô hình nhỏ gọn: **text2att** cho tác vụ hiểu văn bản mô tả (tiếng Anh hoặc tiếng Việt) với mô hình gốc là BERT; **att2midi** cho tác vụ sinh nhạc với mô hình gốc là GPT-2. Với kiến trúc này, ngoài việc dễ dàng lựa chọn mô hình và điều chỉnh tham số huấn luyện phù hợp nhất có thể cho mỗi tác vụ, chúng tôi còn có thể so sánh với các mô hình hiện có một cách thuận tiện. Bên cạnh đó, chúng tôi áp dụng kỹ thuật giảm ma trận LoRA (Low-Rank Adaptation) để tối ưu chi phí, hiệu suất, và đơn giản hoá quá trình huấn luyện 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. Bên cạnh các yếu tố về mô hình và kỹ thuật, chúng tôi cũng thực hiện thu thập dữ liệu âm nhạc và văn bản mô tả âm nhạc, đặc biệt là văn bản tiếng Việt, với web scraping techniques và prompt engineering. Với khoảng 200 triệu tham số, mô hình của chúng tôi cho thấy hiệu quả vượt trội hơn xấp xỉ 12% trên độ đo ASA - Average Sample-wise Accuracy, khi so sánh với mô hình có cùng độ lớn được dựng lại từ original source code của MuseCoCo và huấn luyện trên cùng bộ dữ liệu mà chúng tôi thu thập. 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

Nhiều nghiên cứu trong những năm gần đây về lĩnh vực sinh nhạc từ văn bản mô tả như MuseNet[13] (2019) từ OpenAI, MusicGen[3] (2023) từ Facebook, và

MuseCoCo[11] (2023) từ Microsoft cho thấy bài toán này có hai hướng tiếp cận chính là text-to-audio và text-to-symbolic-music. Hướng tiếp cận audio thường gặp những hạn chế như không có khả năng cho phép chỉnh sửa một cách chi tiết dẫn đến thiếu sự tự do sáng tạo trong việc sử dụng kết quả sinh nhạc vào sáng tác, dữ liệu huấn luyện cồng kềnh, khó khăn trong việc kiểm tra các đặc trưng của âm nhạc (nhạc cụ, time signature, pitch range, etc) trong kết quả đầu ra, etc. Điều đó thôi thúc chúng tôi lựa chọn hướng symbolic music, cụ thể là text-to-midi.

Theo Thomas Lidy et al[10], symbolic music is "music stored in a notation-based format (e.g., MIDI), which contains excplicit information about note onsets and pitch on individual tracks (for different instruments), but in contrast to digital audio no sound". Giải pháp này, với dạng cụ thể là MIDI, cung cấp sự linh hoạt và hiệu quả cho việc chỉnh sửa theo ý muốn của người sử dụng sau khi âm nhạc được tạo ra, kích thước dữ liệu nhỏ hơn, và quá trình trích xuất các đặc trưng của đoạn nhạc dễ dàng hơn so với audio. Điều này cho phép kiểm soát chi tiết và chính xác hơn đối với các thuộc tính âm nhạc, giúp tạo ra các bản nhạc dài hơn và phù hợp hơn với yêu cầu của người dùng một cách dễ dàng, trong khi sử dụng tài nguyên tính toán trong quá trình huấn luyện và quá trình inference ít hơn audio.

Ngoài vấn đề audio-or-symbolic-music, một thách thức phổ biến khi áp dụng các mô hình sinh nhạc đó là việc giúp mô hình hiểu các mô tả theo cảm tính của người dùng, ví dụ: *"Một bài nhạc có tốc độ trung bình và cho cảm giác thư giản"*. Việc chuyển đổi những mô tả mang tính cảm tính này thành các thông số kỹ thuật cụ thể như tempo, key, hay time signature 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 khăn, đặc biệt đối với những người không có kiến thức kỹ thuật sâu về âm nhạc. Vấn đề này tuy được giải quyết ở ở các công trình theo hướng text-to-symbolic-music nhưng lại gặp khó khăn trong việc đa dạng hoá nguồn dữ liệu huấn luyện cho văn bản mô tả.

Để giải quyết các vấn đề trên, nghiên cứu của chúng tôi có những đề phương pháp đề xuất và đóng góp có thể mô tả qua bốn ý chính:

1. Chúng tôi đã sử dụng định dạng tập tin MIDI (dạng tập tin có thể lưu trữ notation-based music, nói cách khác là symbolic music) thay vì audio. Định dạng này có thể chỉnh sửa được và phù hợp với nhiều mục đích khác nhau như thay đổi nhạc cụ, melody, harmony, time signature, etc, của đoạn nhạc. Điều đó làm cho MIDI phù hợp với mục đích sáng tạo âm nhạc.
2. Chúng tôi đã tham khảo kiến trúc hai mô hình huấn luyện riêng biệt cho từng tác vụ của MuseCoCo[11] gồm: hiểu thuộc tính âm nhạc từ văn bản đầu vào và sinh nhạc từ các thuộc tính đó. 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. Từ đó, chúng tôi đã mở rộng ra để phát triển mô hình đa ngôn ngữ gồm: tiếng Anh và tiếng Việt. Bên cạnh đó, chúng tôi đã cải tiến phần sinh nhạc bằng việc áp dụng mô hình ngôn ngữ lớn GPT-2[14] kết hợp với kỹ thuật LoRA[7] để tối ưu hoá tài nguyên và chi phí huấn luyện.

Chỉ với 200 triệu tham số, giải pháp này cho ra kết quả tốt hơn 12% về độ chính xác dựa trên độ đo ASA[11] khi so sánh với mô hình có cùng độ lớn được dựng lại từ original source code của MuseCoCo và huấn luyện trên cùng bộ dữ liệu mà chúng tôi thu thập.

3. Chúng tôi đã đề xuất ra kỹ thuật hậu xử lý dữ liệu để kiểm tra tính đúng đắn của đầu ra của mô hình và sử dụng kỹ thuật nội suy dựa trên quy tắc nhạc lý để sửa lỗi những đoạn nhạc đầu ra bị sai.
4. Chúng tôi đã áp dụng kỹ thuật prompt engineering nhằm tăng cường đa dạng dữ liệu, nhất là dữ liệu tiếng Việt, với nhiều ngữ cảnh khác nhau nhằm bao quát được những ngữ cảnh cụ thể mà người dùng nhập vào.

2 Related Work

Lĩnh vực sinh nhạc từ văn bản, hay text-to-music, đã và đang phát triển mạnh mẽ trong những năm gần đây, với sự xuất hiện của nhiều công trình nghiên cứu đa dạng và phong phú. Các nhà khoa học và kỹ sư đã đưa ra nhiều ý tưởng sáng tạo, áp dụng các kiến trúc và mô hình ngôn ngữ lớn như Transformers[15] và mạng GANs[5] (Generative Adversarial Networks). Những mô hình này giúp biến đổi văn bản thành âm nhạc một cách tự động, mở ra những khả năng mới mẻ và nhiều ứng dụng thực tiễn trong lĩnh vực trong sáng tạo âm nhạc. Một số công trình nổi bật như MuLan - Music Language Model for Audio Generation[8], một mô hình tiên tiến trong lĩnh vực text-to-music với khả năng tạo ra âm nhạc từ mô tả văn bản một cách hiệu quả và sáng tạo. MULAN sử dụng các kỹ thuật học máy và xử lý ngôn ngữ tự nhiên để chuyển đổi các mô tả cảm xúc, phong cách và nhịp điệu từ văn bản thành các đoạn nhạc phức tạp; MusicLM - Generating Music From Text[1], 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; MUGEN - A Playground for Video-Audio-Text Multimodal Understanding and GENeration[6], một 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; and more. Đa số các công trình này đều tập trung vào việc sinh ra audio, chỉ có số ít là thiên về việc sinh ra các dạng symbolic music.

Các công trình trước đây thường trực tiếp tạo ra nhạc từ mô tả văn bản, hay nói cách khác là kiến trúc sử dụng một mô hình duy nhất, dẫn đến việc thiếu sự kiểm soát rõ ràng trong quá trình tạo nhạc. Chúng tôi đã tham khảo kiến trúc của **MuseCoCo - Generating Symbolic Music from Text[11]** để giải quyết vấn đề đó. MuseCoCo[11] sử dụng kiến trúc hai lớp để tạo ra symbolic music từ mô tả văn bản với độ chính xác cao. Nhạc được tạo ra ở định dạng này dễ dàng chỉnh sửa và có thể được kiểm soát rõ ràng thông qua các giá trị thuộc tính được rút trích từ văn bản mô tả đầu vào.

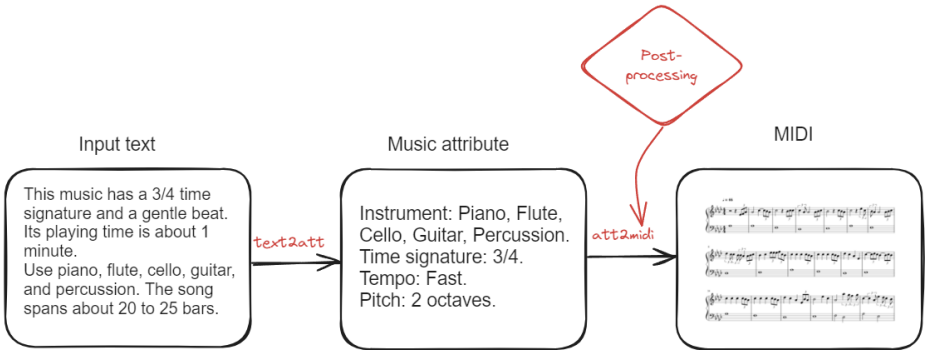

Fig. 1. Kiến trúc tổng quan

3 Proposed Method

3.1 Overview

Kiến trúc chúng tôi sử dụng, được mô tả ở hình 1, có thể chia làm ba giai đoạn chính: giai đoạn 1 - hiểu văn bản mô tả; giai đoạn 2 - sinh nhạc; và giai đoạn 3 - hậu xử lý dữ liệu (nếu dữ liệu bị lỗi).

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 (nhanh, chậm, nhạc cụ được sử dụng, etc) và định lượng (pitch range phân bố trên bao nhiêu octave, đoạn nhạc dài bao nhiêu giây, etc) của đoạn nhạc cần tạo ra. Định dạng dữ liệu cho mô hình ở giai đoạn này là các template tiếng Anh mô tả âm nhạc ở dạng ngôn ngữ tự nhiên và các template tiếng Việt tương ứng. Bên cạnh ngôn ngữ tự nhiên, mỗi template chứa placeholder cho các nhãn định tính và định lượng đã nêu (nếu có). Một ví dụ template tiếng Anh:

"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]."

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. Định dạng dữ liệu cho mô hình ở giai đoạn này là các cặp "source - target", nói cách khác là "command - music". Trong đó, mỗi `command` chứa các cặp metadata abbreviation (xem bảng 1) và giá trị tương ứng; `music` là dữ liệu âm nhạc dạng MIDI được token hoá thành âm nhạc dạng văn bản dựa trên REMI[9] của MuseCoCo[11].

Sau khi hoàn thành quá trình huấn luyện, bước tiếp theo là thực hiện hậu xử lý. Mặc dù mô hình sinh đã được huấn luyện kỹ lưỡng, vẫn luôn tồn tại rủi ro rằng dữ liệu đầu ra có thể không chuyển đổi được về dạng thông tin mong muốn là midi. Do đó, chúng tôi đề xuất một thuật toán kiểm tra và sửa lỗi nhằm đảm bảo rằng kết quả của mô hình có thể chuyển đổi thành dạng dữ liệu đúng.

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 Giai đoạn 1: Mô hình text2att

Chúng tôi dựa vào BERT[4] - một mô hình thuộc loại encoder trong việc trích xuất đặc trưng văn bản, và kiến trúc từ bài báo MusicBERT[17], để xây dựng mô hình dự đoán và phân loại các thuộc tính âm nhạc từ văn bản mô tả (xem hình 2). Chúng tôi phát triển một biến thể với các tuỳ 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 thêm các token [CLS_i] chính là các thông tin của các nhãn đã để cập ở bảng trên. Sau khi đi qua BERT[4] để trích xuất ra các đặc trưng, đầu ra là các logits, sau đó sẽ đi qua một lớp Softmax để phân loại nhãn.

3.3 Giai đoạn 2: Mô hình att2midi

Chúng tôi sử dụng kiến trúc GPT-2[14] được tối ưu bằng cách áp dụng phương pháp LoRA[7] để giảm số lượng tham số cần huấn luyện và tăng tốc quá trình huấn luyện nhưng vẫn giữ được hiệu suất xấp xỉ việc huấn luyện toàn bô tham số. 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, pitch, time signature, bar, tempo, etc, được mã hóa chính xác để mô hình có thể hiểu và học từ dữ liệu (xem hình 3). Bên cạnh đó, chúng tôi dựa vào bộ tokenizer REMI[9] để token hóa dữ liệu, sau đó tiến hành finetune trên tập dữ liệu của mình.

3.4 Giai đoạn 3: Hậu xử lý

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

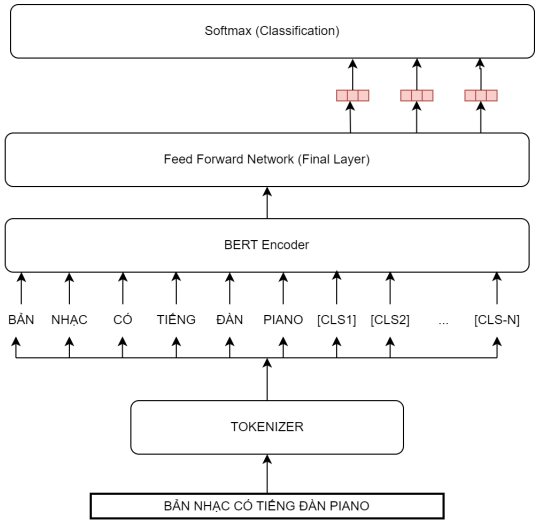

Fig. 2. Kiến trúc mô hình text2att sử dụng BERT

Fig. 3. Mô hình sinh nhạc dạng "command - music"

này được ánh xạ từ bộ quy tắc dựa trên REMI[9] của MuseCoCo[11] (xem quy luật sắp xếp thuộc tính âm nhạc dạng văn bản của MuseCoCo[11] ở bảng 2).

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 4):

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

Table 2. Quy luật sắp xếp các thuộc tính âm nhạc dạng văn bản của MuseCoCo[11]

Abbreviation	Attribute	Next Abbreviation
s	Time Signature	b, o.
o	Position	t.
t	Tempo	i.
i	Instrument	p.
p	Pitch	d.
d	Duration	v.
v	Velocity	i, b, p, o.
b	Bar	s.

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[1].

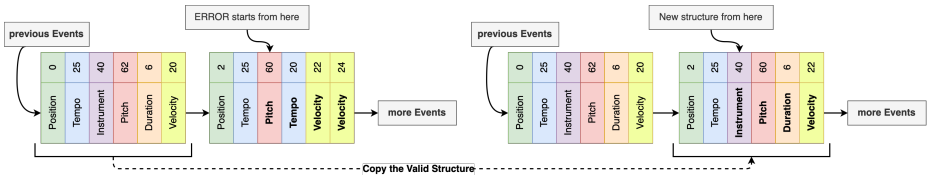

Fig. 4. Ví dụ minh hoạ quá trình hậu xử

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

Datasets - 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[11] tạo ra bằng việc sử dụng ChatGPT[12] trong quá trình nghiên cứu. Ngoài ra, chúng tôi cũng sử dụng prompt engineering với ChatGPT[12] 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 3.

[1] 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).

Table 3. Tên và giá trị tương ứng của các nhãn trong mỗi câu template từ bài báo MuseCoCo[11] (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.

Datasets - 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[11], 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[11] 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. 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[9] của MuseCoCo[11]. Đây sẽ là phần music trong định dạng dữ liệu mục tiêu.

System Configuration - text2att Để huấn luyện giai đoạn text2att. Các siêu tham số được sử dụng trong quá trình huấn luyện mô hình BERT[4] 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 2000, 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[14] sử dụng kỹ thuật LoRA[7] (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[7], 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[14] 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[7] 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[7] 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 Dataset and Metrics Chúng tôi thực hiện đánh giá riêng lẻ trên từng mô hình. Để đánh giá mô hình trên, chúng tôi xây dựng một bộ kiểm tra tiêu chuẩn bao gồm 1000 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 từ prompt đầu vào.

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[11]), 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[16]: 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[16]: 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 việc nghe và cảm nhận.

4.2 Baselines

Trong nghiên cứu này, chúng tôi đã so sánh phương pháp của mình với công trình MuseCoCo[11]. Để công bằng, chúng tôi dựng lại mô hình của MuseCoCo bằng original source code và thiết lập cấu hình hệ thống với độ lớn về số lượng tham số, cấu hình máy chủ huấn luyện, dữ liệu huấn luyện, định dạng dữ liệu đầu ra, etc, tương tự các mô hình chúng tôi đã đề xuất. Kết quả so sánh được tổng hợp sau quá trình huấn luyện và dự đoán với bộ checkpoint tốt nhất.

4.3 Results

Khi đánh giá mô hình **text2att**, 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. Accuracy của từng instrument của thuộc tính Instrument

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. Accuracy của những thuộc tính khác

Attribute	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 chúng tôi thu được một số đánh 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 GPT-2 vượt trội hơn về hầu hết các độ đo. Đặc biệt, trong các độ đo ASA và Rhythm Danceability, LoRA GPT-2 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 GPT-2**	**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 5, 6, 7, và 8.

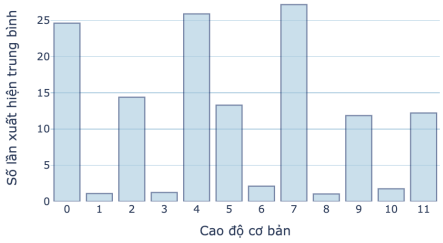

Fig. 5. Số lần xuất hiện trung bình của các giá trị cao độ cơ bản (LoRA GPT-2)

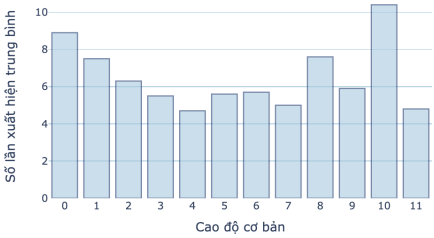

Fig. 6. Số lần xuất hiện trung bình của các giá trị cao độ cơ bản (Kiến trúc MuseCoCo)

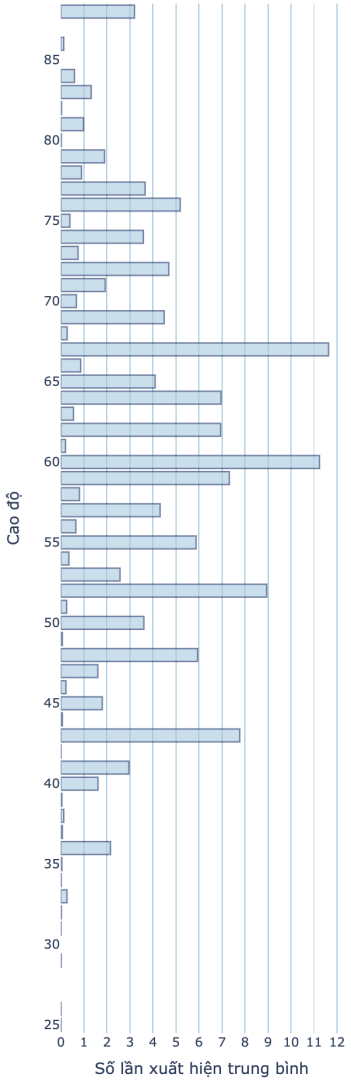

Fig. 7. Số lần xuất hiện trung bình của các giá trị cao độ (LoRA GPT-2)

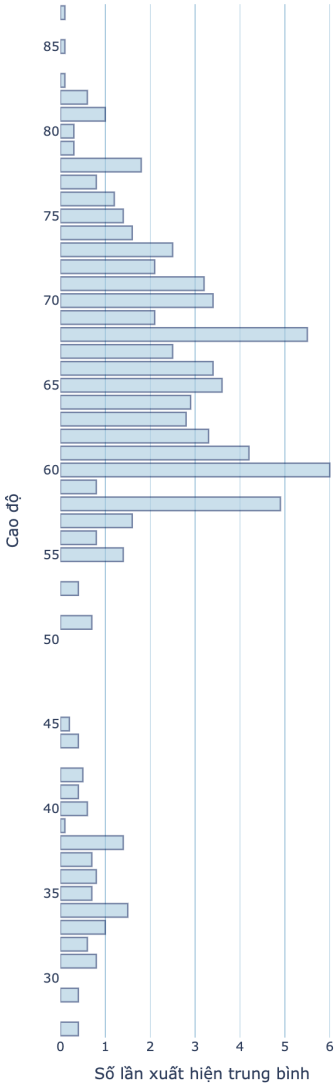

Fig. 8. Số lần xuất hiện trung bình của các giá trị cao độ (Kiến trúc MuseCoCo)

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.

Kết quả nghiên cứu cho thấy việc sử dụng các mô hình decoder như GPT-2[14] rất phù hợp với quá trình sinh nhạc, trong khi mô hình encoder như BERT[4] phù hợp với việc phân loại và trích xuất các đặc trưng. Chúng tôi cũng đã áp dụng kỹ thuật LoRA[7] 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[7] 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. Anderson, C., Carlton, D., Miyakawa, R., Schwachhofer, D.: Hooktheory, https://www.hooktheory.com
3. Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., Défossez, A.: Simple and controllable music generation. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019), https://arxiv.org/abs/1810.04805
5. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014), https://arxiv.org/abs/1406.2661
6. 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)
7. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021), https://arxiv.org/abs/2106.09685
8. Huang, Q., Jansen, A., Lee, J., Ganti, R., Li, J.Y., Ellis, D.P.W.: Mulan: A joint embedding of music audio and natural language (2022), https://arxiv.org/abs/2208.12415

9. Huang, Y.S., Yang, Y.H.: Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions (2020)
10. Lidy, T., Rauber, A.: Music Information Retrieval, pp. 448–456. IGI Global (1 2009). https://doi.org/10.4018/978-1-59904-879-6.ch046, http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-879-6.ch046
11. Lu, P., Xu, X., Kang, C., Yu, B., Xing, C., Tan, X., Bian, J.: Musecoco: Generating symbolic music from text (2023), https://arxiv.org/abs/2306.00110
12. OpenAI: Chatgpt, https://chatgpt.com
13. Payne, Christine: Musenet (2019), https://openai.com/blog/musenet/
14. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023), https://arxiv.org/abs/1706.03762
16. Warnerfjord, M.: Evaluating chatgpt's ability to compose music using the midi file format (2023)
17. 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