

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Phạm Quốc Vương - Ngô Phi Hùng

HỆ THỐNG AI
HỖ TRỢ SÁNG TÁC NHẠC

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng 07/2024

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Phạm Quốc Vương - 20120406

Ngô Phi Hùng - 20120486

HỆ THỐNG AI
HỖ TRỢ SÁNG TÁC NHẠC

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN

TS. Trần Duy Hoàng

Tp. Hồ Chí Minh, tháng 07/2024

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu của tôi dưới sự hướng dẫn của

Thuyết minh chỉnh sửa đề tài

Theo mẫu thuyết minh chỉnh sửa đề tài (nếu có thay đổi tên đề tài, nội dung báo cáo... trong cuốn báo cáo sau bảo vệ)

Nhận xét hướng dẫn

Theo bản nhận xét của giảng viên hướng dẫn (có chữ kí) do giáo vụ cung cấp.

Nhận xét phản biện

Theo bản nhận xét của giảng viên phản biện (có chữ kí) do giáo vụ cung cấp.

Lời cảm ơn

Tôi xin chân thành cảm ơn ...

Đề cương chi tiết

Theo mẫu đề cương đã điền và nộp cho giáo vụ (*phải có chữ kí của giảng viên hướng dẫn*).

Mục lục

Nhận xét của GV hướng dẫn	i
Nhận xét của GV phản biện	iv
Lời cảm ơn	v
Đề cương	vi
Mục lục	vii
Tóm tắt	xi
1 Giới thiệu	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu đề tài	1
1.3 Cách tiếp cận	2
1.4 Đóng góp	4
2 Các công trình liên quan	7
2.1 Cơ sở lý thuyết	7
2.1.1 Lý thuyết âm nhạc	7
2.1.2 Kỹ thuật huấn luyện nền tảng	18
2.1.3 Fairseq	21
2.2 Các nghiên cứu liên quan	22
2.3 Dữ liệu huấn luyện	23

2.3.1	Nguồn dữ liệu	23
2.3.2	Kỹ thuật thu thập	24
2.3.3	Các định dạng dữ liệu chính	24
2.4	Mô hình ngôn ngữ lớn	34
2.4.1	Transformer	35
2.4.2	BERT	36
2.4.3	GPT2	37
2.5	Độ đo hiệu suất mô hình	38
2.5.1	Mô hình trích xuất đặc trưng	38
2.5.2	Mô hình sinh nhạc	39
3	Phương pháp đề xuất	40
3.1	Xác định chi tiết vấn đề	40
3.2	Tổng quan về phương pháp	40
3.2.1	Kiến trúc tổng quát	40
3.2.2	Lí do lựa chọn	42
3.2.3	Các bước thực hiện	44
3.3	Chuẩn bị dữ liệu	45
3.3.1	Dữ liệu ngôn ngữ tự nhiên	45
3.3.2	Dữ liệu âm nhạc	46
3.4	Huấn luyện mô hình	47
3.4.1	Mô hình trích xuất đặc trưng câu văn bản	47
3.4.2	Mô hình sinh nhạc	49
3.5	Hậu xử lý dữ liệu	52
3.6	Chọn phương pháp đánh giá mô hình	52
3.6.1	Các thông số đánh giá của nghiên cứu có trước	52
3.6.2	Các thông số đánh giá nhóm bổ sung	52
4	Kết quả thí nghiệm	53
4.1	Kết quả huấn luyện	53
4.1.1	Mô hình tích xuất đặc trưng	53
4.1.2	Mô hình tích xuất đặc trưng	54

4.1.3	Mô hình tích xuất đặc trưng	55
4.2	Phần mềm demo	55
5	Kết luận	60
	Danh mục công trình của tác giả	62
	Tài liệu tham khảo	63
A	Ngữ pháp tiếng Việt	64
B	Ngữ pháp tiếng Nôm	65

Danh sách hình

1.1	Prototype: Màn hình chat với mô hình	5
1.2	Prototype: Màn hình chỉnh sửa midi được mô hình tạo ra	6
2.1	Ví dụ nhịp $\frac{3}{4}$	11
	13figure.2.2	
2.3	Tổng quan về cơ chế của kỹ thuật LoRA	21
2.4	Ví dụ nốt nhạc được chuyển về dạng tập tin MIDI	27
2.5	Ví dụ nốt nhạc được chuyển về dạng REMI	32
2.6	Một đoạn dữ liệu âm nhạc dạng văn bản của MuseCoco . .	33
3.1	Kiến trúc tổng quát của mô hình	42
4.1	Kiến trúc tổng quát của mô hình	57
4.2	Kiến trúc tổng quát của mô hình	58
4.3	Kiến trúc tổng quát của mô hình	59

Danh sách bảng

2.1	Các hình nốt thường gặp	8
2.2	Các dấu lặng thường gặp	8
2.3	Cách gọi tên 12 nốt nhạc cơ bản	14
2.4	Đặc tả dữ liệu âm nhạc dạng văn bản của MuseCoco . . .	34
3.2	Các hyperparameter được sử dụng trong quá trình huấn luyện mô hình BERT với lớp Softmax	49
3.3	51
3.4	Các hyperparameter được sử dụng trong quá trình huấn luyện mô hình GPT-2 với lớp Softmax và kỹ thuật LoRA .	52
3.5	Các hyperparameter được sử dụng trong quá trình huấn luyện mô hình GPT-2 với lớp Softmax và kỹ thuật LoRA .	52
4.1	Độ chính xác của các thuộc tính âm nhạc	54
4.2	55

Tóm tắt

Sau “cơn sốt” ChatGPT năm 2022, phần mềm phát triển dựa trên trí tuệ nhân tạo (AI) ra đời ngày càng nhiều và trở nên phổ biến, mang đến lợi ích cho hàng loạt lĩnh vực như lập trình, đồ họa, phim ảnh, truyền thông, v.v. Sản xuất âm nhạc trên máy tính cũng không ngoại lệ. Với nhiều kiểu ứng dụng như sinh nhạc theo dạng text-to-audio, text-to-midi¹, AI hỗ trợ phối trộn âm thanh (mixing), v.v, sự kết hợp giữa trí tuệ nhân tạo và sáng tạo âm nhạc mở ra nhiều cơ hội mới và tiềm năng hứa hẹn. Dựa trên xu hướng đó, nhóm chọn thực hiện đề tài “Hệ thống AI hỗ trợ sáng tác nhạc” với hướng “sinh nhạc theo dạng text-to-midi”. Bằng kỹ thuật chủ đạo là Masked Language Modeling và Next Sentence Prediction, nhóm mong muốn kết hợp các nghiên cứu đã có, cũng như kiến thức của bản thân, để cho ra một hệ thống AI hỗ trợ sáng tác nhạc, giúp khơi dậy và duy trì nguồn cảm hứng cho người làm nhạc trên máy tính nói riêng và người sáng tạo âm nhạc nói chung trong quá trình tạo ra các tác phẩm độc đáo và phong phú.

Nội dung “một số kết quả đạt được” thêm vào sau.

¹MIDI - Musical Instrument Digital Interface: tương tự việc con người đọc các ký hiệu trên sheet nhạc, máy tính đọc các ký hiệu đó nhưng được biểu diễn ở dạng MIDI để hiểu được bản nhạc.

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Nêu một số tên thành tựu quan trọng về mô hình hoá âm nhạc từ trước đến nay và tên kỹ thuật áp dụng chính trong các thành tựu đó

Dẫn dắt đến lí do chọn sinh midi mà không phải audio (trên phương diện tiện dụng cho đối tượng hướng đến là người làm nhạc).

Nêu vấn đề text chỉ có mô tả kỹ thuật hoặc có thêm mô tả cảm xúc nhưng chưa thấy được sự áp dụng rõ ràng của các nghiên cứu trước và mong muốn dùng text có mô tả đặc điểm kỹ thuật và đặc điểm cảm xúc của bài nhạc để generate ra midi.

Nêu vấn đề sinh bài nhạc mới theo style bài nhạc cho trước (nếu hoàn thành).

1.2 Mục tiêu đề tài

- Để giúp người sáng tác âm nhạc mở rộng phạm vi sáng tạo và có nguồn cảm hứng không giới hạn, nhóm đánh giá việc tạo ra “hệ thống AI hỗ trợ sáng tác nhạc” là một trong những cách nhanh và hiệu quả nhất.
- Khi thực hiện đề tài, nhóm muốn mang lại được các lợi ích: tăng

cường hiệu suất công việc (thay vì phải dành nhiều thời gian cho việc tạo ra các giai điệu và hoà âm cơ bản, người sáng tác có thể tập trung vào các khâu phức tạp hơn như tô điểm cho giai điệu và hoà âm được công cụ tạo ra để có được thành quả tốt và đúng ý muốn nhất, giành thời gian tiết kiệm được cho khâu phối khí và các khâu phức tạp hơn sau đó), mở rộng khả năng và phạm vi sáng tạo nhờ việc cung cấp các nét nhạc với cách đi giai điệu và nhịp điệu mà người viết nhạc chưa từng nghĩ đến, hỗ trợ người không biết nhạc lý vẫn có thể sáng tác, và nhiều lợi ích khác.

- Đề tài có kết quả tốt sẽ mang ý nghĩa tăng cường sự sáng tạo trong việc sáng tác âm nhạc, thay đổi phương pháp sáng tác âm nhạc truyền thống, nâng cao hiệu suất và hiệu quả của ngành công nghiệp âm nhạc, cũng như góp một phần nhỏ vào lĩnh vực nghiên cứu và phát triển sự kết hợp giữa trí tuệ nhân tạo và âm nhạc.

1.3 Cách tiếp cận

Hướng phát triển nhóm đề xuất:

- Nâng cao tính toàn diện của mô hình bằng việc kết hợp khả năng sinh text-to-midi của MuseCoco và khả năng hiểu mô tả nhạc theo cả phương diện kỹ thuật (giống MuseCoco) lẫn phương diện cảm xúc và cảm nhận của con người hoặc theo ngữ cảnh (giống MusicLM và MuGen).
- Xây dựng mô hình có bộ nhớ (có thể hỏi và trả lời với người dùng, nhớ ngữ cảnh cuộc trò chuyện để đưa ra các câu trả lời tiếp theo).
- Phát triển thành mô hình đa ngôn ngữ thay vì chỉ có tiếng Anh.
- Triển khai mô hình và phát triển một ứng dụng web giúp người dùng có thể dễ dàng tiếp cận.

Phương pháp tiếp cận:

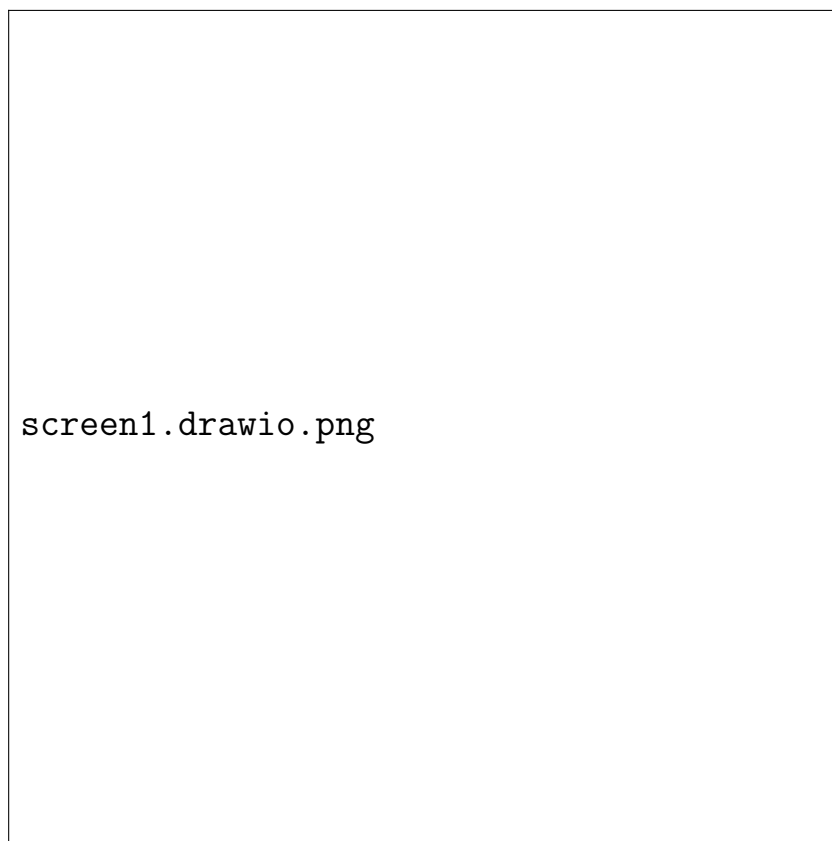
- Tìm hiểu về lĩnh vực âm nhạc: các khái niệm trong âm nhạc như phách, nhịp, cao độ, trường độ, thang âm, v.v; các nguyên tắc hoà âm cơ bản; cách phối hợp các nhạc cụ trong bản nhạc; các bước để sáng tác ra một bản nhạc; v.v.
- Khảo sát, đánh giá cách tiếp cận của các nghiên cứu được đề cập ở phần 2.2.
- Thu thập dữ liệu từ các nguồn đã nêu ở phần 2.3.1: tải xuống thông thường với các bộ dữ liệu từ MuseCoco và MusicCaps; dùng Python để cào dữ liệu từ Hooktheory và kỹ thuật đa luồng để tăng hiệu suất cào.
- Viết công cụ chuyển đổi cấu trúc dữ liệu giữa các mô hình cần thử nghiệm.
- Tiếp cận các mô-hình-được-đào-tạo-trước (BERT, GPT2) để thử nghiệm các tác vụ trích xuất đối tượng và sinh văn bản.
- Nghiên cứu kỹ thuật Name Entity Recognition: Trích xuất từ văn bản người dùng nhập những thuộc tính được định nghĩa trước. Ví dụ: với thuộc tính được định nghĩa trước là “nhạc cụ”, khi người dùng nhập vào “Bản nhạc thư giãn được đệm bằng piano.” thì đầu ra sẽ là “piano”.
- Nghiên cứu kỹ thuật Masked Language Modeling: Đây là quá trình mô hình hóa ngôn ngữ bằng cách che đi ngẫu nhiên các từ trong câu đầu vào, sau đó chạy toàn bộ câu đã được xử lý như trên đi qua mô hình và để mô hình dự đoán các từ đã bị che là gì. Điều này cho phép mô hình học được một biểu diễn hai chiều của câu (từ trái qua phải và từ phải qua trái), từ những thuộc tính đã trích xuất tạo những tiền tố để mô hình có thể sinh ra câu văn bản mô tả nhạc

hoàn chỉnh. Ví dụ: với câu “Tôi là sinh viên Công nghệ Thông tin”, ta huấn luyện mô hình sao cho khi truyền vào mô hình câu đã bị che “Tôi là <MASK> Công nghệ Thông tin”, mô hình phải dự đoán được <MASK> là “sinh viên”.

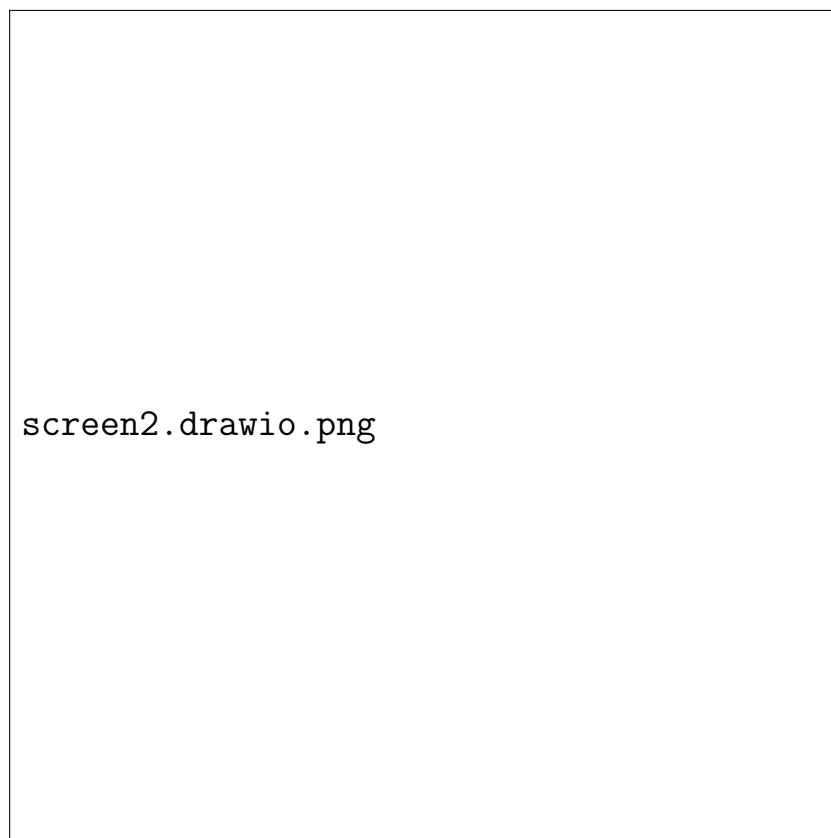
- Nghiên cứu kỹ thuật Next Sentence Prediction: Với đầu vào là một chuỗi các token tiền tố, mô hình phải dự đoán được chuỗi các token hậu tố cho chuỗi token tiền tố đó là gì. Bài toán điển hình cho kỹ thuật này là bài toán sinh thơ - mô hình nhận câu thơ đầu vào và sinh ra câu thơ tiếp theo cho câu thơ mà mô hình nhận được.
- Kết hợp các nghiên cứu có sẵn và ba kỹ thuật trên để cho ra mô hình hiểu được văn bản mô tả nhạc theo cả khía cạnh kỹ thuật và cảm xúc, cảm nhận tự nhiên của con người.
- Nghiên cứu phương pháp tối ưu và nâng cao chất lượng nhạc sinh ra (ở dạng midi) của các mô hình từ những nghiên cứu đã nêu dựa trên cách tiếp cận có sẵn của các nghiên cứu đó và hai kỹ thuật Masked Language Modeling và Next Sentence Prediction.
- Viết công cụ chuyển thông tin từ định dạng đầu ra của mô hình thành dạng midi.
- Xây dựng ứng dụng web.

1.4 Đóng góp

- Mô hình sinh nhạc bằng cách nhận đầu vào là văn bản mô tả đặc điểm bản nhạc bằng ngôn ngữ tự nhiên (tiếng Anh hoặc tiếng Việt).
- Ứng dụng web/Phần mềm sinh ra nhạc theo yêu cầu.
- Prototype (xem Hình 1.1 và Hình 1.2):



Hình 1.1: Prototype: Màn hình chat với mô hình



Hình 1.2: Prototype: Màn hình chỉnh sửa midi được mô hình tạo ra

Chương 2

Các công trình liên quan

- Chương trình này trình bày cơ sở lý thuyết, lý luận khoa học, phân tích các nghiên cứu từ các bài báo tham khảo, các vấn đề về mô hình ngôn ngữ lớn cần được cải tiến trong đó.
- Các phương pháp nghiên cứu được sử dụng trong các đề tài này thuộc nhiều lĩnh vực khác nhau như học máy, học sâu, ngôn ngữ lớn, kết hợp các phương pháp kỹ thuật kỹ thuật như LoRA dùng để hoàn thiện các mô hình ngôn ngữ lớn, kỹ thuật kỹ thuật sinh dữ liệu; thử nghiệm mục tiêu hợp lý để sử dụng một cách có hiệu quả.







2.1 Cơ sở lý thuyết

2.1.1 Lý thuyết âm nhạc







Lý thuyết âm nhạc là nền tảng cho việc hiểu và sáng tạo âm nhạc. Nó bao gồm các khái niệm, nguyên tắc giúp người học nắm bắt và sử dụng các yếu tố âm nhạc một cách hiệu quả. Trong mục này, chúng tôi tập trung vào những lý thuyết cơ bản nhất, cần thiết nhất và thường xuyên sử dụng trong luận văn, nhằm giúp người đọc có cái nhìn tổng quát, giảm sự phức tạp trong quá trình hiểu các ý tưởng, cũng như không bị quá tải thông tin trong quá trình đọc.

2.1.1.1 Hình nốt, dấu lặng, và trường độ

Khi hát một bài hát bất kỳ, ta luôn rơi vào một trong hai trường hợp: hát theo lời bài hát, hoặc tạm ngưng ở vị trí không có lời. Để biểu diễn nốt nhạc được hát ở vị trí có lời, ta dùng hình nốt (xem các hình nốt thường gặp ở bảng 2.1). Để biểu diễn các vị trí tạm ngưng hát, ta dùng dấu lặng (xem các dấu lặng thường gặp ở bảng 2.2). Mỗi nốt hoặc dấu lặng được hát hoặc nghỉ trong thời gian bao lâu được xác định bằng trường độ tương ứng của hình nốt hoặc dấu lặng đó.

Tên hình nốt	Tên tiếng Anh	Cách viết khác	Ký hiệu
Nốt tròn	Whole note		
Nốt trắng	Half note	$\frac{1}{2}$ note	
Nốt đen	Quarter note	$\frac{1}{4}$ note	
Nốt móc đơn	Eighth note	$\frac{1}{8}$ note	
Nốt móc đôi	Sixteenth note	$\frac{1}{16}$ note	
Nốt móc ba	Thirty-second note	$\frac{1}{32}$ note	

Bảng 2.1: Các hình nốt thường gặp

Tên dấu lặng	Tên tiếng Anh	Cách viết khác	Ký hiệu
Dấu lặng tròn	Whole rest		
Dấu lặng trắng	Half rest	$\frac{1}{2}$ rest	
Dấu lặng đen	Quarter rest	$\frac{1}{4}$ rest	
Dấu lặng móc đơn	Eighth rest	$\frac{1}{8}$ rest	
Dấu lặng móc đôi	Sixteenth rest	$\frac{1}{16}$ rest	
Dấu lặng móc ba	Thirty-second rest	$\frac{1}{32}$ rest	

Bảng 2.2: Các dấu lặng thường gặp

Mối tương quan trường độ giữa các hình nốt:

$$\circ = 2\text{J} = 4\text{J} = 8\text{J} = 16\text{J} = 32\text{J}$$

Mối tương quan trường độ giữa các dấu lặng:

$$\text{—} = 2\text{—} = 4\text{ } \text{ } = 8\text{ } \text{ } = 16\text{ } \text{ } = 32\text{ } \text{ }$$

Các hình nốt và dấu lặng có hậu tố giống nhau thì có trường độ bằng nhau. Ví dụ: nốt tròn có trường độ bằng dấu lặng tròn, nốt đen có trường độ bằng dấu lặng đen, v.v. Và quan trọng nhất, việc so sánh trường độ giữa các hình nốt và dấu lặng chỉ có ý nghĩa khi chúng nằm trong một hoặc các bản nhạc có cùng loại nhịp (time signature, xem mục 2.1.1.2) và cùng ký hiệu tốc độ (tempo, xem mục 2.1.1.3).

Ngoài các ký hiệu hình nốt và dấu lặng trên còn có dấu chấm đôi đặt bên phải hình nốt, làm tăng trường độ của hình nốt hoặc dấu lặng đó thêm $\frac{1}{2}$. Ví dụ:

$$\circ. = \circ + \text{J} = 3\text{J}$$

Việc đặt liên tiếp các dấu chấm đôi bên phải một hình nốt hoặc dấu lặng sẽ làm tăng trường độ của hình nốt hoặc dấu lặng đó theo công thức:

$$duration(n) = \frac{d}{1} + \frac{d}{2} + \frac{d}{4} + \dots + \frac{d}{2^n},$$

với *duration* là trường độ, *n* là số dấu chấm đôi, *d* là trường độ nốt gốc. Ví dụ:

$$\circ.. = \circ + \text{J} + \text{J} = 7\text{J}$$

2.1.1.2 Ô nhịp và các loại nhịp

Ô nhịp (measure, hay còn gọi là bar) là các cụm nốt nhạc (và dấu lặng nếu có) được chia ra từ các nốt (và dấu lặng) ghi theo thứ tự trên sheet nhạc¹. Mỗi ô nhịp được chia thành nhiều phách, mỗi phách có trường độ được quy định dựa trên trường độ của phách đơn vị.

Thông tin trường độ phách đơn vị và độ dài mỗi ô nhịp được suy ra từ ký hiệu nhịp (time signature) của bài nhạc. Ký hiệu nhịp thường được đặt ở phần đầu của sheet nhạc, và có thể thay đổi trong suốt bản nhạc. Ký hiệu nhịp bao gồm hai con số, một số ở trên và một số ở dưới, giống như một phân số², với ý nghĩa:

- Số ở trên (numerator): Độ dài ô nhịp tính theo số *phách đơn vị*³.
- Số ở dưới (denominator): Trường độ của *phách đơn vị*.

Ví dụ với một nhịp thông dụng - nhịp $\frac{3}{4}$ (xem hình 2.1), số 3 ở trên cho biết mỗi ô nhịp có độ dài là 3 phách đơn vị, số 4 ở dưới cho biết mỗi phách đơn vị có trường độ bằng trường độ nốt $\frac{1}{4}$ (nốt đen). Ngoài nhịp $\frac{3}{4}$, còn có các nhịp thông dụng khác như $\frac{2}{4}$, $\frac{4}{4}$, $\frac{6}{8}$, v.v; hoặc các nhịp ít phổ biến hơn như $\frac{9}{8}$, $\frac{12}{8}$, v.v.

¹Sheet nhạc là một loại văn bản bài nhạc. Trong đó, các dòng kẻ nhạc và nốt nhạc mô tả giai điệu, hoà âm, v.v, của bản nhạc là nội dung chính.

²Chỉ có điểm giống về cách ký hiệu, không áp dụng phép toán rút gọn như phân số.

³“*Phách đơn vị*” là thuật ngữ âm nhạc tạm thời do người viết đề xuất để phân biệt với thuật ngữ “*phách*” trong luận văn (xem chi tiết ở đoạn cuối của mục 2.1.1.2) do các thuật ngữ liên quan đến vấn đề này còn gây nhập nhằng cho người mới. Thuật ngữ tạm thời này không tương đương với thuật ngữ “*beat unit*” (trong trường hợp người đọc tìm kiếm thông tin trong các tài liệu tiếng Anh).



3_4_TimeSig.png

Hình 2.1: Ví dụ nhịp $\frac{3}{4}$

Những thông tin được trình bày ở trên đã đủ để sử dụng cho việc lập trình tính toán trong luận văn. Tuy nhiên, *phách* và *phách đơn vị* của một nhịp là hai cụm từ thường bị nhầm lẫn, nên để tránh hai thuật ngữ này bị hiểu sai hoặc hiểu phiến diện, chúng tôi xin phép gọi mở thêm một số thông tin ở đoạn kế tiếp để người đọc có thể tìm hiểu khi có nhu cầu.

Trong lý thuyết âm nhạc, số ở trên của ký hiệu nhịp chỉ có ý nghĩa chỉ ra độ dài ô nhịp theo số *phách đơn vị*, không có ý nghĩa chỉ ô nhịp có bao nhiêu *phách*; số ở dưới chỉ có ý nghĩa chỉ ra trường độ mỗi *phách* đơn vị là hình nốt nào, không có ý nghĩa chỉ *phách* có trường độ là hình nốt nào. Với nhịp $\frac{3}{4}$, ta thấy rất đơn giản khi số phách và số phách đơn vị trong một ô nhịp đều là 3, mỗi phách và phách đơn vị đều có trường độ là nốt $\frac{1}{4}$, chúng thống nhất với nhau. Với trường hợp nhịp $\frac{6}{8}$, tuy số phách đơn vị trong mỗi ô nhịp là 6, mỗi phách đơn vị là nốt $\frac{1}{8}$, nhưng số phách trong mỗi ô nhịp lại là 2, mỗi phách có trường độ bằng 3 nốt $\frac{1}{8}$. Hoặc với nhịp $\frac{7}{8}$, trong

khi số phách đơn vị trong ô nhịp và trường độ phách đơn vị thống nhất với ký hiệu nhịp, số phách trong ô nhịp lại là 3, trường độ của mỗi phách lần lượt là 3 nốt $\frac{1}{8}$, 3 nốt $\frac{1}{8}$, 2 nốt $\frac{1}{8}$, hoặc bất kỳ sự hoán đổi vị trí nào giữa 3 cụm nốt khác nhau về trường độ này. Để có thêm nhiều thông tin, người đọc có thể tìm kiếm với các từ khoá: “*nhịp đơn và nhịp kép*”, “*simple vs. compound time signatures*”, “*division vs. subdivision in music*”, v.v.

2.1.1.3 Tốc độ bản nhạc

Tốc độ của bản nhạc (tempo) xác định tốc độ của bản nhạc, được đo bằng số lượng phách trong mỗi phút (BPM - beats per minute). Trên sheet nhạc, tốc độ được ký hiệu theo quy tắc:

$$\text{Hình nốt lấy làm chuẩn} = \text{Số phách mỗi phút.}$$

Hình nốt lấy làm chuẩn có thể là *phách* hoặc *phách đơn vị* của nhịp của bản nhạc (xem mục 2.1.1.2 để phân biệt hai khái niệm này) tùy theo loại nhịp. Ví dụ:

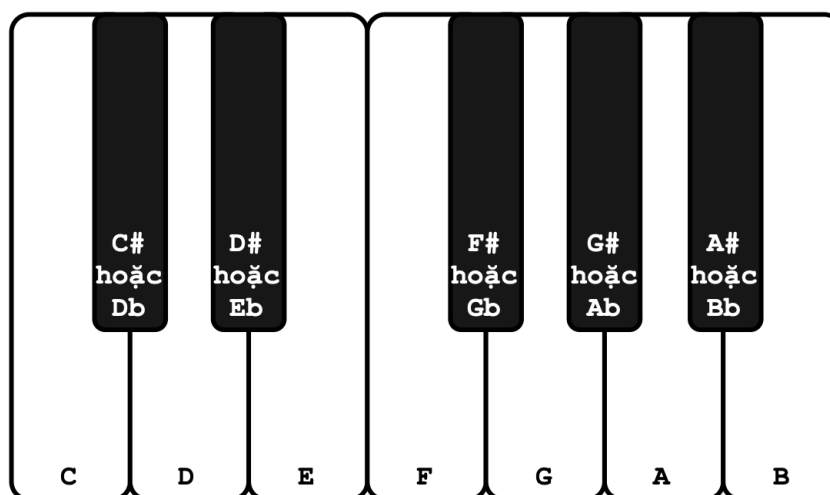
- Một bản nhạc nhịp $\frac{4}{4}$, ký hiệu tốc độ $\text{♩} = 128$, sẽ có 128 nhịp mỗi phút, mỗi nhịp tương ứng với một nốt $\frac{1}{4}$.
- Một bản nhạc nhịp $\frac{6}{8}$, tốc độ 100BPM, có hai cách ký hiệu nhịp: hoặc $\text{♩} = 100$ với ý nghĩa bài nhạc có tốc độ 100 nhịp mỗi phút, mỗi nhịp tương ứng với 1 nốt đen chấm đôi (bằng 3 nốt $\frac{1}{8}$) (tạm gọi là *cách 1 - dùng phách*); hoặc $\text{♪} = 100$ với ý nghĩa bài nhạc có tốc độ 100 nhịp mỗi phút, mỗi nhịp tương ứng với 1 nốt $\frac{1}{8}$ (tạm gọi là *cách 2 - dùng phách đơn vị*).

Về bản chất (xem mục 2.1.1.2, đoạn cuối, về *phách* và *phách đơn vị*), với nhịp $\frac{6}{8}$, cách ký hiệu thứ nhất là phù hợp hơn. Tuy nhiên, hiện nay, nhiều phần mềm làm việc với âm nhạc cho phép sử dụng cả hai cách ký hiệu như phần mềm soạn thảo sheet nhạc MuseScore 4 (soạn thảo âm nhạc bằng các ký hiệu âm nhạc), hoặc hoạt động mặc định theo cách ký hiệu thứ hai như

phần mềm sản xuất âm nhạc FL Studio 21 (soạn thảo âm nhạc dựa trên MIDI). Trong luận văn này, chúng tôi chọn lập trình dựa trên cách ký hiệu thứ hai với lí do cách này có sự tương đồng cao với các số trên ký hiệu nhịp, tạo sự thuận tiện cho việc tính toán với các loại nhịp phức tạp.

2.1.1.4 Cao độ

Khi một câu nhạc được hát lên, ngoài diễn đạt mỗi từ trong lời bài hát cần ngân dài bao lâu - tương ứng với trường độ của nốt nhạc (xem mục 2.1.1.1), người hát còn diễn đạt mỗi từ đó được hát ở những nốt cao thấp thế nào - tương ứng với cao độ của nốt nhạc. Ở các cấp giáo dục phổ thông, chúng ta được biết 7 nốt nhạc cơ bản: Đô, Rê, Mi, Fa, Sol, La, và Si, ký hiệu bằng chữ cái tiếng Anh lần lượt là C, D, E, F, G, A, và B, tương ứng với các phím màu trắng của đàn piano. Tuy nhiên, trên piano còn có các phím màu đen (xem hình minh hoạ 2.2). Đây là lúc ta cần đến ký hiệu dấu thăng (\sharp , tiếng Anh là sharp) và dấu giáng (\flat , tiếng Anh là flat) để có thể gọi tên các nốt nhạc tương ứng với các phím màu đen này.



Hình 2.2: Hình minh hoạ 12 nốt nhạc cơ bản với đàn piano⁴

Hai phím đàn trắng hoặc đen bất kỳ nằm kế nhau trên đàn piano có

⁴Trên các loại đàn phím (keyboard) như piano, những cụm phím theo bố cục này được xếp liên tiếp nhau để tạo thành dãy phím của đàn.

khoảng cách cao độ là $\frac{1}{2}$ cung⁵ (hay còn gọi là nửa cung⁶). Bằng cách thêm vào bên phải của tên nốt nhạc một dấu thăng với tác dụng tăng cao độ lên nửa cung, hoặc dấu giáng để giảm cao độ xuống nửa cung, ta có thể gọi tên các nốt đen một cách dễ dàng. Xem cách gọi tên 12 nốt nhạc cơ bản ở bảng 2.3.

Ký hiệu	Cách đọc tiếng Anh	Cách đọc tiếng Việt
C	C	Đô
C \sharp (hoặc D \flat)	C sharp (hoặc D flat)	Đô thăng (hoặc Rê giáng)
D	D	Rê
D \sharp (hoặc E \flat)	D sharp (hoặc E flat)	Rê thăng (hoặc Mi giáng)
E	E	Mi
F	F	Fa
F \sharp (hoặc G \flat)	F sharp (hoặc G flat)	Fa thăng (hoặc Sol giáng)
G	G	Sol
G \sharp (hoặc A \flat)	G sharp (hoặc A flat)	Sol thăng (hoặc La giáng)
A	A	La
A \sharp (hoặc B \flat)	A sharp (hoặc B flat)	La thăng (hoặc Si giáng)
B	B	Si

Bảng 2.3: Cách gọi tên 12 nốt nhạc cơ bản

Ở bảng trên, về mặt lý thuyết, các nốt C, E, F, và B có tồn tại cách ký hiệu biểu diễn bằng nốt kẻ nó và dấu hoá, ví dụ: C \flat tương đương với B, E \sharp tương đương với F, v.v. Tuy nhiên, việc áp dụng cách ký hiệu như ví dụ cho bốn nốt trên rất ít khi được sử dụng, do gây ra một số

⁵Cung - whole tone hoặc whole step, là đại lượng để tính khoảng cách cao độ giữa hai nốt nhạc.

⁶Trong thuật ngữ tiếng Anh, nửa cung được gọi là semitone hoặc half step.

vấn đề phức tạp khi đọc và chơi bản nhạc (sẽ được giải thích ở mục 2.1.1.5 về thang âm), nên chúng tôi không liệt kê để băng sát với thực tế.

2.1.1.5 Thang âm

Khi một người đang hát, nếu đột nhiên người nghe cảm thấy nốt nhạc nào đó có cao độ không hoà quyện và chệch hẳn khỏi nhạc nền, ta thường nói người hát đang “hát sai tone”, “hát bị chênh cao độ”, hoặc “hát bị ngang”. Các nốt gây ra cảm giác không đúng đó cho người nghe chính là những nốt bị lệch khỏi thang âm của bài nhạc.

TODO

Thang âm (scale) là một dãy các nốt nhạc theo một trật tự cao độ xác định. Thang âm cơ bản và phổ biến nhất là thang âm trưởng (Major scale) và thang âm thứ (Minor scale). Một thang âm trưởng bao gồm các nốt nhạc theo thứ tự: cung – cung – nửa cung – cung – cung – cung – nửa cung. Ví dụ, thang âm Đồ trưởng (C Major) gồm các nốt: Đồ, Rê, Mi, Fa, Sol, La, Si, Đồ.

Thang âm thứ có ba loại chính: Thứ tự nhiên (Natural Minor), Thứ hòa thanh (Harmonic Minor), và Thứ giai điệu (Melodic Minor). Thang âm thứ tự nhiên bao gồm các nốt nhạc theo thứ tự: cung – nửa cung – cung – cung – nửa cung – cung – cung. Ví dụ, thang âm La thứ (A Natural Minor) gồm các nốt: La, Si, Đô, Rê, Mi, Fa, Sol, La.

2.1.1.6 Quãng nhạc

Quãng nhạc (interval, gọi tắt là quãng) là khoảng cách cao độ giữa hai nốt nhạc giữa hai cao độ. Các quãng cơ bản bao gồm:

- Quãng 1 (Unison): Cùng một cao độ.
- Quãng 2 (Second): Khoảng cách 1 cung (tone) hoặc 1/2 cung (semi-tone).
- Quãng 3 (Third): Khoảng cách 2 cung hoặc 1,5 cung.

- Quãng 4 (Fourth): Khoảng cách 2,5 cung.
- Quãng 5 (Fifth): Khoảng cách 3,5 cung.
- Quãng 6 (Sixth): Khoảng cách 4,5 cung hoặc 4 cung.
- Quãng 7 (Seventh): Khoảng cách 5,5 cung hoặc 5 cung.
- Quãng 8 (Octave): Khoảng cách 6 cung, nốt cùng tên ở cao độ gấp đôi.

Các quãng nhạc có thể được mô tả thêm bằng các tính từ như trưởng (Major), thứ (Minor), đúng (Perfect), tăng (Augmented), và giảm (Diminished).

2.1.1.7 Các loại hợp âm hai và ba nốt

Hợp âm (chord) là sự kết hợp của ba hoặc nhiều nốt nhạc được vang lên đồng thời. Các hợp âm cơ bản nhất là hợp âm hai nốt (dyad) và hợp âm ba nốt (triad). Hợp âm ba nốt bao gồm ba loại chính:

- Hợp âm trưởng (Major chord): Bao gồm nốt gốc (root), quãng ba trưởng (major third), và quãng năm đúng (perfect fifth). Ví dụ, hợp âm Đồ trưởng (C Major) gồm các nốt: Đồ (C), Mi (E), Sol (G).
- Hợp âm thứ (Minor chord): Bao gồm nốt gốc, quãng ba thứ (minor third), và quãng năm đúng. Ví dụ, hợp âm La thứ (A Minor) gồm các nốt: La (A), Đô (C), Mi (E).
- Hợp âm giảm (Diminished chord): Bao gồm nốt gốc, quãng ba thứ, và quãng năm giảm (diminished fifth). Ví dụ, hợp âm Si giảm (B Diminished) gồm các nốt: Si (B), Rê (D), Fa (F).
- Hợp âm tăng (Augmented chord): Bao gồm nốt gốc, quãng ba trưởng, và quãng năm tăng (augmented fifth). Ví dụ, hợp âm Đồ tăng (C Augmented) gồm các nốt: Đồ (C), Mi (E), Sol (G).

2.1.1.8 Các loại hợp âm từ bốn nốt trở lên

Hợp âm từ bốn nốt trở lên phức tạp hơn và tạo ra các âm sắc đa dạng hơn. Các hợp âm này bao gồm:

- Hợp âm bảy (Seventh chord): Bao gồm bốn nốt nhạc. Các loại phổ biến:
 - Hợp âm bảy trưởng (Major Seventh): Nốt gốc, quãng ba trưởng, quãng năm đúng, và quãng bảy trưởng. Ví dụ, Đồ bảy trưởng (Cmaj7) gồm Đồ (C), Mi (E), Sol (G), Si (B).
 - Hợp âm bảy thứ (Minor Seventh): Nốt gốc, quãng ba thứ, quãng năm đúng, và quãng bảy thứ. Ví dụ, La bảy thứ (Am7) gồm La (A), Đô (C), Mi (E), Sol (G).
 - Hợp âm bảy át (Dominant Seventh): Nốt gốc, quãng ba trưởng, quãng năm đúng, và quãng bảy thứ. Ví dụ, Sol bảy át (G7) gồm Sol (G), Si (B), Rê (D), Fa (F).
 - Hợp âm bảy giảm (Diminished Seventh): Nốt gốc, quãng ba thứ, quãng năm giảm, và quãng bảy giảm. Ví dụ, Si bảy giảm (Bdim7) gồm Si (B), Rê (D), Fa (F), Lab (Ab).
- Hợp âm chín (Ninth chord): Bao gồm năm nốt nhạc, thêm nốt thứ chín vào hợp âm bảy.
- Hợp âm mười một (Eleventh chord): Bao gồm sáu nốt nhạc, thêm nốt thứ mười một vào hợp âm chín.
- Hợp âm mười ba (Thirteenth chord): Bao gồm bảy nốt nhạc, thêm nốt thứ mười ba vào hợp âm mười một.

Các hợp âm phức tạp này thường được sử dụng trong nhạc jazz và các thể loại nhạc khác cần sự phong phú về âm sắc và cảm xúc.

2.1.2 Kỹ thuật huấn luyện nền tảng

2.1.2.1 Masked Language Modelling

Masked Language Model (MLM) là một phương pháp huấn luyện mô hình ngôn ngữ, nổi bật với việc sử dụng trong mô hình BERT của Google. MLM hoạt động bằng cách che đi một phần các từ trong câu đầu vào và yêu cầu mô hình dự đoán các từ bị che giấu đó dựa trên ngữ cảnh xung quanh. MLM buộc mô hình phải học cách biểu diễn từ ngữ theo ngữ cảnh của chúng, thay vì chỉ dựa vào ý nghĩa riêng lẻ của từng từ. Điều này giúp mô hình nắm bắt được các mối quan hệ phức tạp giữa các từ trong câu và xây dựng một biểu diễn ngôn ngữ phong phú hơn.

2.1.2.2 Casual Language Model

Casual Language Modelling (CLM) là một kỹ thuật xử lý trong lĩnh vực ngôn ngữ tự nhiên, đây là một dạng dùng trong các autoregressive model và dùng để dự đoán từ tiếp theo dựa trên chuỗi từ trước đó, thông thường thì CLM sẽ được sử dụng trong các mô hình như GPT 2, GPT 3 dành cho các tác vụ sinh văn bản hay tóm tắt. Lấy một ví dụ thì giả sử cung cấp cho mô hình câu hướng dẫn (prompt) là “hôm nay trời ...” thì nó sẽ dự đoán từ tiếp theo có thể xuất hiện là “nắng”, “mưa”.

Casual Language Modelling hoạt động dựa trên cách phân tích trên một lượng dữ liệu lớn văn bản để rút ra các mẫu có sẵn hay các quy luật trong ngôn ngữ, nó có thể phát hiện ra các cặp từ vựng thường đi cùng nhau. Từ đó có thể sinh ra dữ liệu văn bản một cách hợp lý nhất.

Trong khóa luận này, dữ liệu âm nhạc cũng tuân theo một quy luật nhất định, thì qua đó, chúng tôi đã tận dụng CLM để tạo ra một mô hình sinh nhạc hiệu quả. Khi triển khai CLM thì nó được huấn luyện bằng cách đưa vào một lượng dữ liệu lớn văn bản để mô hình có thể học được cấu trúc ngữ pháp và ngữ nghĩa của ngôn ngữ.

2.1.2.3 LoRA

LoRA ra đời vào năm 2021 do đội ngũ Microsoft Research phát hành thông qua bài báo LoRA: Low-Rank Adaptation of Large Language Models, sau khi các mô hình ngôn ngữ lớn ra đời thì LoRA được đề xuất như một phương pháp hiệu quả để tinh chỉnh các mô hình ngôn ngữ lớn bằng cách tận dụng Low-rank. Từ khi giới thiệu thì LoRA đã nhanh chóng trở thành một kỹ thuật phổ biến để tinh chỉnh các mô hình ngôn ngữ lớn.

Điều làm LoRA trở nên nổi bật chính là hiệu quả về chi phí mà không làm giảm hiệu suất đáng kể bởi thông thường việc huấn luyện một mô hình ngôn ngữ lớn là rất tốn kém chi phí bởi nó đòi hỏi nhiều tài nguyên. Với sự gia tăng chóng mặt của các mô hình thì LoRA được kì vọng giúp có thể dễ dàng tiếp cận các mô hình này.

LoRA tận dụng khái niệm low-rank matrices (ma trận cấp thấp) để huấn luyện và xử lý mô hình hiệu quả và nhanh chóng hơn. Thông thường các mô hình ngôn ngữ lớn thì tốn nhiều bộ nhớ về GPU như GPT 3 với 175 tỷ tham số, hay các mô hình Llama 70 tỷ tham số, việc huấn luyện và tinh chỉnh các mô hình ngôn ngữ này khá tốn kém và có thể tiêu tốn hàng triệu đô la với các mô hình có kích thước như GPT.

Không giống như việc tinh chỉnh truyền thống toàn bộ mô hình thì LoRA chỉ tập trung vào việc huấn luyện lại tập con tham số trên các lớp nhất định, từ đó giảm chi phí bộ nhớ. LoRA tận dụng các ma trận cấp thấp bởi theo ta được biết thì ma trận cấp thấp có thể phân tích được thành tích hai ma trận nhỏ hơn. Điều này giúp biểu diễn được các thông tin phức tạp với ít tham số hơn. Vậy nên có thể huấn luyện các ma trận này và điều chỉnh trên một tác vụ cụ thể mà không cần phải cập nhật lại toàn bộ mô hình.

Đi sâu vào cơ chế của LoRA thì thông thường sẽ có các bước gồm

- Phân rã ma trận có trọng số lớn. Các mô hình như GPT sử dụng ma trận trọng số khổng lồ để lưu trữ các tham số. Kỹ thuật LoRA

cho phép phân rã các ma trận này thành các ma trận nhỏ hơn thông qua kỹ thuật phân rã ma trận cấp thấp, giúp xấp xỉ một ma trận lớn bằng tích của hai ma trận nhỏ hơn. Điều này làm giảm đáng kể số lượng tham số cần huấn luyện, thường chỉ khoảng 1

- Huấn luyện ma trận cấp thấp, tất cả các ma trận cấp thấp mới được thêm vào đều được huấn luyện, giúp quá trình nhanh chóng và hiệu quả.
- Dự đoán kết quả với các bộ trọng số cộng thêm. Sau khi huấn luyện, LoRA không thay đổi mô hình gốc. Trong quá trình huấn luyện, trọng số mới sẽ được cộng thêm vào trọng số ban đầu của mô hình.

Ưu điểm khi sử dụng LoRA để tinh chỉnh và huấn luyện mô hình ngoài tối ưu tài nguyên ra còn vì ma trận cấp thấp có kích thước nhỏ nên việc thêm chúng vào mô hình ban đầu sẽ không gây ra độ trễ. Có thể chuyển đổi linh hoạt giữa các tác vụ bởi do ma trận cấp thấp có kích thước nhỏ nên chúng có thể dễ dàng thay đổi linh hoạt khi cần thiết giữa các tác vụ và người dùng khác nhau. Nên từ điều này có thể tạo ra một mô hình đa chức năng đáp ứng nhiều nhu cầu của người dùng mà tiết kiệm với nguồn tài nguyên.

Khi áp dụng LoRA thì chúng tôi đã sử dụng thư viện PEFT (Parameter-Efficient Fine-Tuning) - một phương pháp nhằm mục tiêu giảm số lượng tham số cần cập nhật trong quá trình tinh chỉnh mà vẫn duy trì hiệu suất cao.



Hình 2.3: Tổng quan về cơ chế của kỹ thuật LoRA

2.1.3 Fairseq

Fairseq, một bộ công cụ học sâu phát triển bởi Facebook AI Research (FAIR), được thiết kế đặc biệt để hỗ trợ nghiên cứu và thử nghiệm các mô hình sequence-to-sequence, đóng một vai trò quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Fairseq cung cấp một khung làm việc hiệu quả và linh hoạt cho việc triển khai, huấn luyện và thử nghiệm các mô hình NLP tiên tiến như mô hình dịch máy tự động, nhận dạng giọng nói, tóm tắt văn bản và tạo chú thích hình ảnh. Được trang bị các tính năng như huấn luyện song song trên nhiều GPU, hỗ trợ đa ngôn ngữ, và tối ưu hóa các quá trình huấn luyện với kỹ thuật mới nhất, Fairseq cho phép các nhà nghiên cứu và nhà phát triển nâng cao chất lượng và hiệu quả của các mô hình học sâu.

Trong các ứng dụng thực tế, Fairseq đã chứng minh khả năng đáp ứng

với các yêu cầu khắt khe của các hệ thống NLP hiện đại. Từ việc dịch ngôn ngữ tự động với độ chính xác cao đến phát triển các hệ thống nhận dạng giọng nói tiên tiến và tạo ra bản tóm tắt văn bản súc tích, Fairseq đã giúp đẩy nhanh quá trình nghiên cứu và triển khai các giải pháp NLP hiệu quả. Điều này không chỉ củng cố vị thế của FAIR như một trung tâm nghiên cứu AI hàng đầu, mà còn tăng cường khả năng cạnh tranh của Facebook trong lĩnh vực công nghệ AI toàn cầu.

2.2 Các nghiên cứu liên quan

- **MuseCoco - Generating Symbolic Music from Text**[7]: 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”.
- **MusicLM: Generating Music From Text**[4]: 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 synth⁷ 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.
- **MUGEN: A Playground for Video-Audio-Text Multimodal Understanding and GENeration**[5]: MuGen là mô hình được nghiên cứu để hiểu và sinh âm thanh cho video game dựa trên đầu

⁷Còn gọi là lead synthesizer - một loại nhạc cụ điện tử dùng để đệm các tuyến giai điệu chính của bản nhạc.

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.”.

Nhận xét về các nghiên cứu nêu trên:

- Tuy đều là các nghiên cứu sinh nhạc từ văn bản, nhưng MusicLM và MUGEN lại sinh nhạc ở định dạng âm thanh (text-to-audio), khác với hướng sinh nhạc ở dạng midi (text-to-midi) mà nhóm lựa chọn.
- MuseCoco tuy sinh nhạc ở dạng midi, phù hợp với hướng nghiên cứu của nhóm nhưng phần văn bản mô tả nhạc còn giới hạn trong việc mô tả các đặc tính kỹ thuật (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.”), không mô tả theo cảm xúc và cảm nhận tự nhiên của con người như MusicLM (ví dụ: “Bản nhạc chậm và buồn, với giai điệu du dương được chơi bằng guitar và có piano đệm hợp âm cho giai điệu đó.”) hay sinh nhạc theo kịch bản, mô tả ngữ cảnh như MuGen.
- Hầu hết các phương pháp tiếp cận cận chỉ thực hiện qua các bản demo nhỏ hoặc chưa phát triển thành một ứng dụng để giúp người dùng phổ thông có thể tiếp cận được.

2.3 Dữ liệu huấn luyện

2.3.1 Nguồn dữ liệu

- Hai bộ liệu huấn luyện của bài báo **MuseCoco - Generating Symbolic Music from Text**[7]. Bộ thứ nhất gồm các cặp “Câu mô tả các tính chất của bản nhạc bằng ngôn ngữ tự nhiên - Câu mô tả được mã hoá (encode) theo định dạng xác định trước”. Bộ thứ hai gồm các dòng dữ liệu chứa thông tin các đoạn nhạc được mã hoá (encode) theo định dạng được xác định trước.

- Bộ dữ liệu MusicCaps⁸ từ bài báo **MusicLM: Generating Music From Text**[4] với các cặp “Khoá xác định bản nhạc trên YouTube - Đoạn văn bản mô tả các đặc điểm của bản nhạc” là dữ liệu chính.
- Dữ liệu do nhóm thu thập từ **Hooktheory**⁹ - 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. Dữ liệu nhóm thu thập có thông tin chính gồm các cặp “Thông tin ký âm và hoà âm của một bản nhạc (có thể chuyển về dạng midi) - Đoạn văn bản nhận xét về bản nhạc và một số chỉ số đánh giá bản nhạc”.

2.3.2 Kỹ thuật thu thập

2.3.3 Các định dạng dữ liệu chính

2.3.3.1 MIDI

MIDI (Musical Instrument Digital Interface) là một giao thức kỹ thuật số được phát triển vào đầu thập niên 1980 để cho phép các nhạc cụ điện tử, máy tính và các thiết bị âm thanh khác giao tiếp và kiểm soát lẫn nhau. MIDI không truyền âm thanh thực sự, mà truyền các sự kiện âm nhạc như nốt nhạc (note), tốc độ nhấn/lực nhấn phím đàn (velocity), và các thông số điều khiển khác.

Các thành phần cơ bản của MIDI gồm:

1. Giao thức truyền dữ liệu: MIDI sử dụng giao thức truyền dữ liệu theo kiểu nối tiếp với tốc độ 31.25 kbps. Các thông điệp MIDI gồm các byte trạng thái (status byte) và byte dữ liệu (data byte).
2. Thông điệp MIDI:

⁸Liên kết đến bộ dữ liệu tại Kaggle: <https://www.kaggle.com/datasets/googleai/musiccaps>, truy cập lần cuối 01/04/2024.

⁹Liên kết đến trang web: <https://www.hooktheory.com>, truy cập lần cuối 01/04/2024.

- Note On/Off: Thông điệp này cho biết khi nào một nốt nhạc bắt đầu phát (on) hay dừng (off), và có thể bao gồm thông tin về lực nhấn phím đàn (velocity).
 - Control Change: Được sử dụng để thay đổi các thông số như âm lượng, độ ngân (sustain), hay các điều khiển khác.
 - Program Change: Thay đổi âm sắc hay chương trình của một nhạc cụ.
 - Pitch Bend: Thay đổi cao độ của nốt nhạc.
 - System Messages: Các thông điệp dùng để đồng bộ hóa các thiết bị, khởi động hoặc dừng playback, và các chức năng hệ thống khác.
3. Kết nối: Gồm các cổng kết nối cho phép truyền thông điệp MIDI giữa các thiết bị:
- MIDI In: Nhận dữ liệu.
 - MIDI Out: Gửi dữ liệu.
 - MIDI Thru: Chuyển tiếp dữ liệu từ “MIDI In” đến các thiết bị khác.

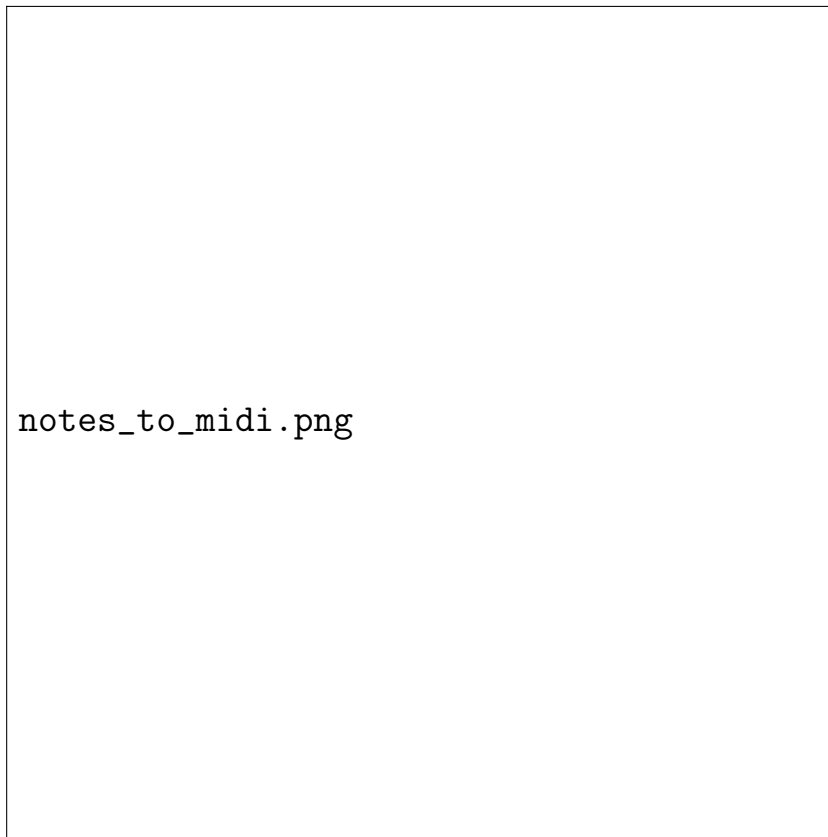
Lợi ích mà MIDI mang lại (nhất là với quá trình làm việc với âm nhạc số):

- Tính tương thích: MIDI là một tiêu chuẩn quốc tế được áp dụng rộng rãi, nên các thiết bị từ các nhà sản xuất khác nhau có thể làm việc cùng nhau dễ dàng.
- Tính linh hoạt: MIDI cho phép điều khiển và chỉnh sửa âm nhạc một cách dễ dàng, từ việc thay đổi âm sắc đến chỉnh sửa các ghi chú nhạc.
- Hiệu quả: Dữ liệu MIDI rất nhỏ gọn, nên việc truyền và lưu trữ rất hiệu quả.

Ngày nay, MIDI trở thành tiêu chuẩn trong nhiều phần mềm, thiết bị liên quan đến âm nhạc với đa dạng ứng dụng như:

- Sản xuất âm nhạc: MIDI cho phép các nhạc sĩ và nhà sản xuất kết nối và điều khiển nhiều thiết bị âm thanh, từ đàn tổng hợp âm thanh (synthesizer), máy chơi trống (drum machine), đến phần mềm âm nhạc trên máy tính.
- Biểu diễn trực tiếp: MIDI được sử dụng để đồng bộ hóa các thiết bị trên sân khấu, điều khiển ánh sáng, và các hiệu ứng đặc biệt.
- Học tập và giảng dạy âm nhạc: MIDI hỗ trợ các phần mềm giáo dục âm nhạc, cho phép học sinh thực hành và cải thiện kỹ năng chơi nhạc cụ.

Trong khoá luận này, chúng tôi sử dụng định dạng tập tin MIDI - một trong các thể hiện của giao thức trên, và các kiểu dữ liệu được biến đổi dựa trên định dạng vừa nêu. Loại tập tin này chứa dữ liệu MIDI, cho phép lưu trữ và phát lại các sự kiện âm nhạc đã được ghi lại, giúp người dùng chia sẻ, chỉnh sửa, và phát lại các bản nhạc trên nhiều thiết bị và phần mềm khác nhau một cách dễ dàng.



Hình 2.4: Ví dụ nốt nhạc được chuyển về dạng tập tin MIDI

Ngoài lí do về tính tiện dụng của MIDI cho đối tượng người dùng hướng đến là người làm nhạc trên máy tính đã nêu ở phần giới thiệu, chúng tôi chọn định dạng này thay vì tập tin âm thanh vì các nguyên nhân chi tiết sau:

- **Tính cấu trúc cao:**
 - **Ưu điểm của tập tin MIDI:** Tập tin MIDI chứa thông tin về các nốt nhạc, độ dài, lực nhấn phím đàn, và các điều khiển khác. Điều này cung cấp cho mô hình một cấu trúc rõ ràng và có tổ chức, giúp dễ dàng phân tích, học tập, và chỉnh sửa. Ngoài ra, tập tin MIDI hoàn toàn có thể chuyển thành dữ liệu âm thanh.
 - **Điểm yếu của tập tin âm thanh:** Tập tin âm thanh chỉ chứa sóng âm thanh liên tục, không có thông tin trực tiếp về cấu trúc

âm nhạc như nốt nhạc hay nhịp điệu. Mô hình phải trích xuất các tính năng này từ dữ liệu sóng âm, một quá trình phức tạp và dễ bị lỗi. Hơn nữa, tập tin âm thanh không thể chỉnh sửa sâu đến từng nốt nhạc như MIDI, dẫn đến hệ quả nhạc sinh ra không thể tùy chỉnh sâu theo ý muốn của người dùng.

- **Kích thước dữ liệu nhỏ gọn:**

- **Ưu điểm của tập tin MIDI:** Tập tin MIDI có kích thước rất nhỏ so với tập tin âm thanh, giúp tiết kiệm không gian lưu trữ và giảm chi phí xử lý dữ liệu.
- **Điểm yếu của tập tin âm thanh:** Tập tin âm thanh có kích thước lớn, đặc biệt là các tập tin chất lượng cao (ví dụ: wav, flac), dẫn đến việc tiêu tốn nhiều không gian lưu trữ và tài nguyên xử lý khi huấn luyện mô hình.

- **Tính đa năng:**

- **Ưu điểm của tập tin MIDI:** Tập tin MIDI là một tiêu chuẩn quốc tế và có thể được sử dụng trên nhiều nền tảng và thiết bị khác nhau mà không cần chuyển đổi định dạng.
- **Điểm yếu của tập tin âm thanh:** Các định dạng âm thanh có thể khác nhau (mp3, wav, flac) và đôi khi không tương thích với một số phần mềm hoặc thiết bị, đòi hỏi phải chuyển đổi định dạng, gây mất thời gian và có thể giảm chất lượng.

- **Không bị ảnh hưởng bởi tín hiệu nhiễu và tạp âm:**

- **Ưu điểm của tập tin MIDI:** Tập tin MIDI không chứa các tạp âm hay nhiễu âm thanh như trong tập tin âm thanh, giúp mô hình tập trung vào các yếu tố âm nhạc chính xác hơn.
- **Điểm yếu của tập tin âm thanh:** Tập tin âm thanh thường chứa nhiễu và tạp âm từ môi trường ghi âm, đòi hỏi các bước

xử lý và làm sạch dữ liệu phức tạp trước khi có thể sử dụng cho huấn luyện mô hình.

- **Khả năng phân tích và hiểu biết cao:**
 - **Ưu điểm của tập tin MIDI:** Dữ liệu MIDI dễ dàng phân tích hơn vì các thông tin về nốt nhạc, nhịp điệu và hòa âm đã được xác định rõ ràng.
 - **Điểm yếu của tập tin âm thanh:** Tập tin âm thanh yêu cầu các kỹ thuật phức tạp để trích xuất thông tin nhạc lý, như nhận diện nốt nhạc, tách nhạc cụ và phân tích hòa âm, các bước này thường tốn nhiều thời gian và dễ gặp sai sót.

2.3.3.2 Lớp MidiFile của thư viện miditoolkit trong Python

Vì tập tin MIDI có cấu trúc tương đối phức tạp, nhiều thư viện Python được ra đời để đơn giản hoá các thao tác trên loại tập tin này, tiêu biểu là `miditoolkit`¹⁰. Trong thư viện trên, lớp `MidiFile` là cấu trúc dữ liệu trọng tâm. Lớp này cho phép lập trình viên đọc, sửa đổi và ghi lại tập tin MIDI một cách dễ dàng, linh hoạt. Trong phạm vi khoá luận, chúng tôi tận dụng các thuộc tính sau của lớp `MidiFile` để làm việc:

1. `ticks_per_beat (int)`: Số tick (đơn vị thời gian trong tập tin MIDI) tương ứng với mỗi nhịp (đơn vị thời gian trong lý thuyết âm nhạc).
2. `tempo_changes (list[TempoChange])`: Danh sách các lần thay đổi tốc độ bản nhạc (`TempoChange`) theo thời gian trong tập tin MIDI. Trong đó `TempoChange` gồm các thuộc tính:

- `tempo (int)`: Tốc độ bản nhạc.

¹⁰Liên kết đến thư viện `miditoolkit` trong Python:
<https://pypi.org/project/miditoolkit/0.1.17/>, truy cập lần cuối 03/06/2024.

- `time (int)`: Thời điểm thay đổi tốc độ bản nhạc; tính theo đơn vị tick.
3. `time_signature_changes (list[TimeSignature])`: Danh sách các lần thay đổi nhịp (`TimeSignature`) của bản nhạc theo thời gian trong tập tin MIDI. Trong đó `TimeSignature` gồm các thuộc tính:
- `numerator (int)`: Số đơn vị nhịp trong một ô nhịp.
 - `denominator (int)`: Độ dài (trường độ) của một đơn vị nhịp.
 - `time (int)`: Thời điểm thay đổi nhịp của bản nhạc; tính theo đơn vị tick.
4. `instruments (list[Instrument])`: Danh sách các đối tượng `Instrument`, mỗi đối tượng này đại diện cho một nhạc cụ trong tập tin MIDI, chứa các nốt nhạc, và gồm các thuộc tính chính sau:
- `program (int)`: Số nguyên xác định số chương trình của nhạc cụ theo chuẩn **General MIDI**¹¹, có giá trị từ 0 đến 127 (tương ứng với các nhạc cụ từ số chương trình 1 đến 128 trong chuẩn được nêu).
 - `is_drum (bool)`: Cho biết nhạc cụ có thuộc nhóm các loại trống hay không.
 - `name (str)`: Tên của nhạc cụ.
 - `notes (list[Note])`: Danh sách các đối tượng `Note`. Mỗi đối tượng này đại diện cho một nốt nhạc, gồm các thuộc tính:
 - `pitch (int)`: Cao độ của nốt nhạc theo chuẩn MIDI, có

¹¹Chi tiết danh sách nhạc cụ ở tài liệu do **MIDI Association** cung cấp: **RP-003_General_MIDI_System_Level_1_Specification_96-1-4_0.1.pdf**, trang 25, mục **General MIDI Sound Set (Table 2)**, tại liên kết <https://midi.org/general-midi-level-1>, truy cập lần cuối 03/06/2024.

giá trị từ 0 đến 127¹², trong đó `pitch` = 60 tương ứng với nốt C4 (middle C) trên đàn piano.

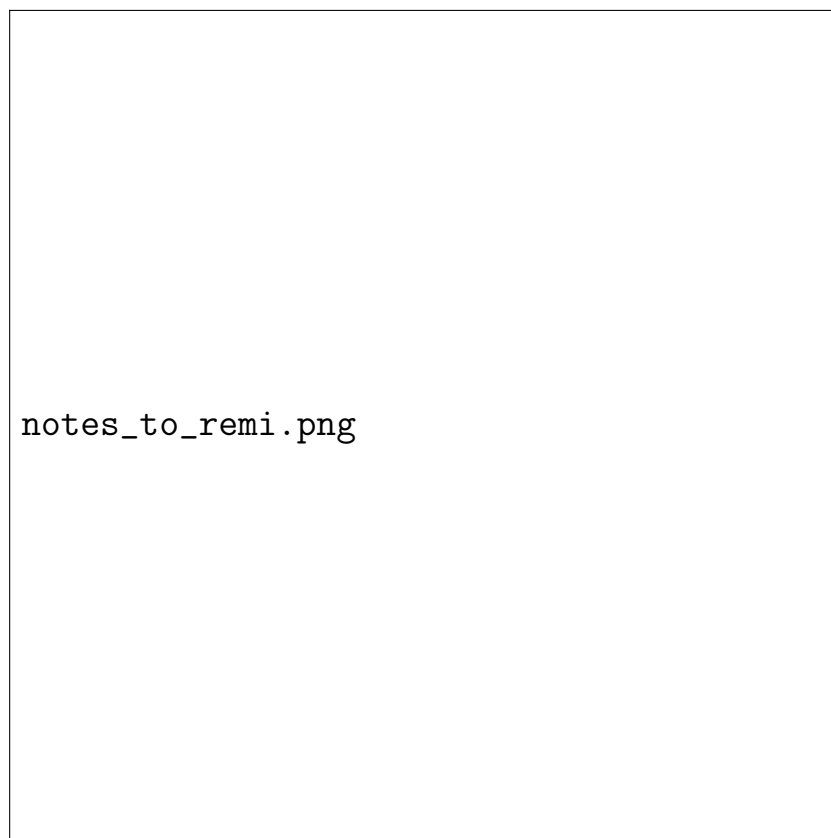
- **velocity**: Tốc độ nhấn/lực nhấn phím đàn, có giá trị từ 0 đến 127.
- **start**: Thời gian bắt đầu của nốt nhạc; tính theo đơn vị tick.
- **end**: Thời gian kết thúc của nốt nhạc; tính theo đơn vị tick.

2.3.3.3 REMI

REMI - REvamped MIDI-derived events, là dạng biểu diễn sự kiện MIDI được bài báo “Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions”[6] đề xuất vào năm 2020 để chuyển đổi các bản nhạc MIDI thành những token rời rạc tương tự văn bản.

So với các kiểu biểu diễn âm nhạc dựa trên MIDI từng áp dụng trong những mô hình sinh nhạc bằng Transformer trước đó, REMI cung cấp cho các mô hình dạng chuỗi (sequence model) một ngữ cảnh nhịp điệu (metrical context) bằng cách thêm thông tin vạch nhịp giữa các ô nhịp của bản nhạc, thay vì chỉ có thông tin thời gian nốt nhạc bắt đầu phát và ngừng phát (note on/note off), để phân chia rõ ràng các ô nhịp, giúp mô-hình mô-hình-hoá các mẫu nhịp điệu (rhythmic pattern - một chuỗi nhịp điệu được lặp lại theo một thứ tự cụ thể) tốt hơn. Từ đó giúp mô hình cho ra các bản nhạc ổn định về mặt nhịp điệu hơn so với các mô hình học thông tin nốt nhạc bắt đầu phát và dừng phát chỉ dựa vào thời gian.

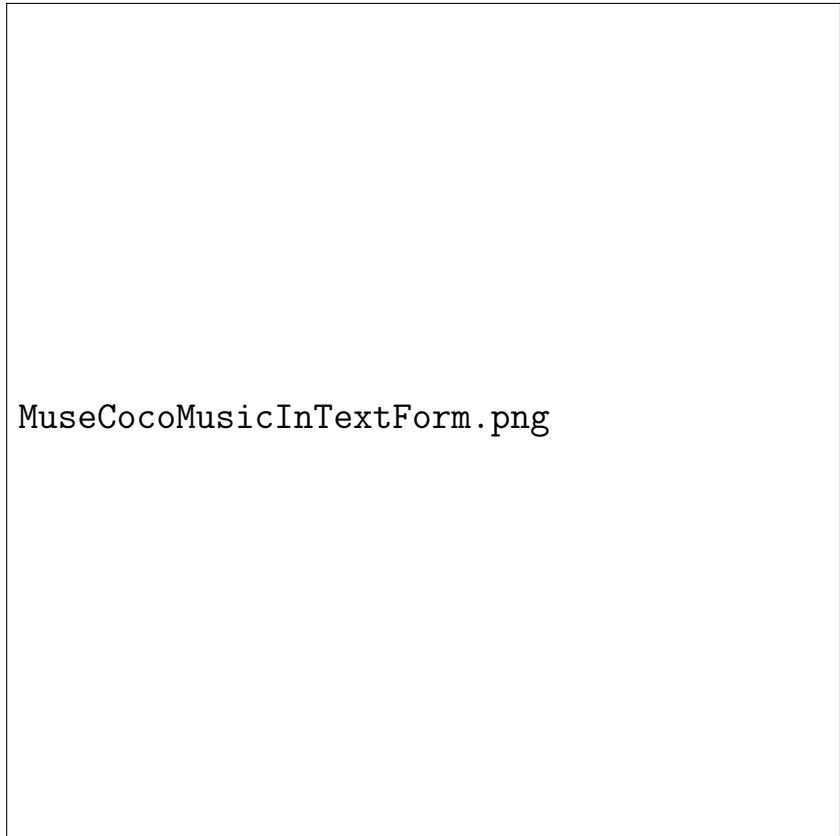
¹²Một số giá trị tương ứng giữa `pitch` và tên nốt nhạc ở tài liệu do **MIDI Association** cung cấp: **General MIDI Level 2 07-2-6 1.2a.pdf**, trang 32, mục **8. Appendix B: GM 2 Percussion Sound Set**, cột **NOTE#**, tại liên kết <https://midi.org/general-midi-level-2-2>, truy cập lần cuối 03/06/2024.



Hình 2.5: Ví dụ nốt nhạc được chuyển về dạng REMI

2.3.3.4 Dữ liệu âm nhạc dạng văn bản của MuseCoco

Bài báo **MuseCoco - Generating Symbolic Music from Text**[7] dựa vào định dạng REMI (đã nêu ở mục 2.3.3.3) để tạo ra định dạng văn bản âm nhạc riêng, từ đó xây dựng bộ dữ liệu âm nhạc từ nhiều bộ dữ liệu MIDI theo định dạng đó (xem ví dụ ở hình 2.6).



`MuseCocoMusicInTextForm.png`

Hình 2.6: Một đoạn dữ liệu âm nhạc dạng văn bản của MuseCoco

Mỗi cặp “`thuộc_tính - giá_trị`” trong định dạng trên phân cách nhau bởi một khoảng trắng; `thuộc_tính` và `giá_trị` phân cách nhau bằng dấu gạch nối; `giá_trị` được chuẩn hoá (normalize) từ các giá trị MIDI tương ứng trong lớp `MidiFile` (đã nêu ở mục 2.3.3.2) bằng bộ chuẩn hoá của bài báo MuseCoco[7]. Xem đặc tả dữ liệu ở bảng 2.4.

Thuộc tính	Tên đầy đủ	Mô tả	Thuộc tính liên 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.

Bảng 2.4: Đặc tả dữ liệu âm nhạc dạng văn bản của MuseCoco

2.3.3.5 Dữ liệu ngôn ngữ tự nhiên mô tả bản nhạc

2.4 Mô hình ngôn ngữ lớn

Trong khoá luận này, nhóm chúng tôi đã áp dụng và thử nghiệm trên các mô hình ngôn ngữ lớn, bao gồm Transformer gốc, BERT và GPT-2. Mục tiêu của chúng tôi là khám phá và đánh giá khả năng và hiệu quả của từng mô hình trong việc xử lý và hiểu ngôn ngữ tự nhiên, cũng như khả năng ứng dụng của chúng trong các tác vụ NLP cụ thể. Transformer, một kiến trúc mạng nơ-ron được sử dụng rộng rãi, là nền tảng cho cả BERT và GPT-2. BERT được thiết kế để hiểu ngữ cảnh hai chiều của một từ trong văn bản, trong khi GPT-2 được tối ưu hóa cho các tác vụ sinh văn bản. Bằng cách triển khai và thử nghiệm các mô hình này trên một loạt các tập dữ liệu và trong các tình huống khác nhau, chúng tôi mong muốn đưa ra những hiểu biết sâu sắc về cách các tiến bộ trong công nghệ ngôn ngữ máy tính có thể được tận dụng để cải thiện và tối ưu hóa chúng. Chúng ta sẽ tìm hiểu kỹ hơn ở phần bên dưới.

2.4.1 Transformer

Transformer là một kiến trúc mạng nơ-ron được giới thiệu vào năm 2017 bởi đội nghiên cứu của Google. Kiến trúc này đã tạo ra một bước ngoặt trong lĩnh vực xử lý ngôn ngữ tự nhiên, đạt được những tiến bộ đáng kể trong nhiều tác vụ khác nhau. Không giống như các mô hình tuần tự truyền thống như RNN và LSTM, vốn xử lý các token tuần tự khiến mô hình huấn luyện chậm và hạn chế trong việc biểu diễn sự phụ thuộc giữa các từ xa nhau trong một câu, Transformer xử lý tất cả các token song song, triệt để giải quyết vấn đề này. Transformer cũng là nền tảng cho các mô hình hiện đại như GPT và BERT.

Đi sâu hơn về kiến trúc Transformer thì ta có các đặc điểm nổi bật tạo nên một Transformer như ngày nay:

- Thứ nhất, Transformer không sử dụng các kiến trúc hồi quy như RNN hay LSTM mà Transformer sử dụng cơ chế self-attention. Đây là thành phần cốt lõi giúp Transformer tạo ra được sự khác biệt. Nó có phép biểu diễn mối quan hệ giữa một với với các từ còn lại. Có nhiều cơ chế attention khác nhau như self-attention, multi-head attention, masked self-attention.
- Bộ mã hóa (Encoder): Bộ mã hóa bao gồm nhiều lớp, mỗi lớp gồm hai thành phần chính là self-attention và mạng nơ-ron truyền thẳng (Feed Forward Network). Ngoài ra, còn có các lớp kết nối tắt (Residual Connection) và chuẩn hóa theo batch (Batch Normalization). Bộ mã hóa được thiết kế để xử lý đầu vào và tạo ra biểu diễn thông tin.
- Bộ giải mã (Decoder): Bộ giải mã bao gồm nhiều lớp tương tự như bộ mã hóa, nhưng có thêm một lớp multi-head attention để chú ý đến đầu ra của bộ mã hóa (Encoder). Bộ giải mã chịu trách nhiệm tạo ra đầu ra mong muốn.
- Mã hóa vị trí: Vì Transformer không có khái niệm về thứ tự của các từ, nên cần một cách để biểu diễn vị trí của mỗi từ trong câu. Điều

này được thực hiện bằng cách thêm một vectơ mã hóa vị trí vào biểu diễn của mỗi từ.

Để hiểu rõ hơn vì sao Transformer ưu việt hơn so với các kiến trúc trước đó thì bởi vì. Thứ nhất, Transformer đã cho ra kiến trúc để có thể tính toán song song cho tất cả các từ trong câu, giúp tăng tốc độ huấn luyện và suy luận. Thứ hai, Cơ chế attention cho phép Transformer nắm bắt được mối quan hệ giữa các từ và hiểu được ngữ cảnh của từng từ, từ đó hiểu được bối cảnh trong câu.

Transformer đã đạt được hiệu suất vượt trội so với các mô hình trước đó trong nhiều tác vụ xử lý ngôn ngữ tự nhiên. Transformer ra đời giúp ra giải quyết được nhiều tác vụ như dịch máy, tóm tắt văn bản, vv. Trong khóa luận này chúng tôi cũng đã áp dụng những kế thừa của Transformer để xây dựng nên hệ thống sản xuất âm nhạc.

2.4.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình được Google phát triển và ra mắt vào năm 2018, đem lại sự đột phá trong lĩnh vực AI, đặc biệt là xử lý ngôn ngữ tự nhiên (NLP). Về sau, BERT còn được ứng dụng trong các lĩnh vực liên quan đến xử lý hình ảnh và âm thanh, mở rộng tầm ảnh hưởng của nó trong thế giới AI.

Kiến trúc của BERT dựa trên mô hình Transformer, sử dụng hoàn toàn các bộ mã hóa (encoder) của Transformer mà không bao gồm phần giải mã (decoder). Kiến trúc này bao gồm nhiều lớp encoder, mỗi lớp gồm các thành phần chính sau:

- **Multihead Self Attention:** Cơ chế này cho phép mô hình chú ý đến các từ khác trong câu khi đang xử lý một từ cụ thể, giúp hiểu rõ bối cảnh và ngữ cảnh của từ.
- **Feed Forward Neural Network:** Kết quả từ lớp self-attention sẽ được đi qua mạng FFN để tiếp tục xử lý. Mỗi lớp encoder có một mạng

FFN riêng biệt.

- Layer Normalization và Residual Connections: Những cơ chế này giúp mô hình ổn định trong quá trình huấn luyện, tăng khả năng học tập và giảm thiểu vấn đề biến mất gradient.
- Pooler: Phần này nằm ở cuối mô hình và được sử dụng để tổng hợp thông tin từ các token đầu ra của encoder, qua đó biểu diễn toàn bộ câu dựa trên các biểu diễn này.

BERT có khả năng thu thập và hiểu ngữ cảnh một cách đa chiều, không chỉ dựa vào ngữ cảnh trước mà còn ngữ cảnh sau của một từ, một cải tiến đáng kể so với các mô hình NLP trước đó chỉ học một chiều.

Trong quá trình huấn luyện, BERT tập trung vào hai mục tiêu chính: Masked Language Model - Phương pháp này dự đoán các token bị che dựa trên ngữ cảnh xung quanh, giúp mô hình hiểu được mối liên kết và ảnh hưởng của các từ lên nhau và Next Sentence Prediction - Kỹ thuật này dự đoán mối liên hệ giữa hai câu, giúp cải thiện khả năng hiểu ngữ cảnh toàn câu và mối liên kết giữa các câu trong văn bản.

BERT đã đạt được thành tựu vượt trội so với các mô hình trước đó trong nhiều tác vụ NLP, từ phân tích cảm xúc đến trả lời câu hỏi, và tiếp tục là một nền tảng quan trọng trong nghiên cứu và ứng dụng AI hiện nay.

2.4.3 GPT2

Một trong những công nghệ đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên, đạt được những thành tựu nổi tiếng và gây tiếng vang lớn cả thế giới đó là GPT3.5, GPT4 hay GPTo. Đặt nền móng cho những mô hình trên đó là GPT 2 do OpenAI phát triển vào năm 2019. Thời điểm GPT 2 ra đời mang tính cải tiến mạnh mẽ so với các mô hình trước đó, và cũng mang tiềm năng để đạt được các thành tựu như ngày. Được mệnh danh là "ma thuật" do khả năng sinh văn bản tự nhiên đầy ấn tượng, GPT-2 dựa

trên kiến trúc Transformer, nổi tiếng với khả năng học sâu và phân tích ngôn ngữ một cách toàn diện.

GPT-2 dựa trên kiến trúc của Transformer. Có nhiều phiên bản với số lượng tham số khác nhau từ 124 triệu đến 1.5 tỷ tham số, nó có quy mô vượt trội so với các mô hình trước đó, với lượng tham số lớn như vậy thì mô hình mở ra tiềm năng to lớn và khả năng ứng dụng trên các tác vụ đặc biệt. Từ đó mô hình có thể áp dụng cho những nhiệm vụ cụ thể, đặc biệt là sinh ra văn bản tự nhiên với độ chính xác cao.

GPT-2 được huấn luyện trên một tập dữ liệu khổng lồ từ internet, bao gồm nhiều loại văn bản khác nhau. Quá trình pre-training giúp mô hình học được cấu trúc ngữ pháp, ngữ nghĩa và các mối quan hệ ngữ cảnh. Sau đó, mô hình có thể được fine-tune trên các tập dữ liệu nhỏ hơn và chuyên biệt cho các nhiệm vụ cụ thể như dịch máy, tóm tắt văn bản hoặc trả lời câu hỏi, hay sinh nhạc.

GPT-2 đòi hỏi rất nhiều tài nguyên tính toán để huấn luyện và triển khai, điều này có thể là một rào cản đối với một số tổ chức và cá nhân.

GPT-2 đã mở đường cho sự ra đời của các mô hình ngôn ngữ lớn hơn và mạnh mẽ hơn như GPT-3 và GPT-4. Những mô hình này tiếp tục cải tiến về quy mô và hiệu suất, mang lại nhiều ứng dụng và khả năng mới cho lĩnh vực xử lý ngôn ngữ tự nhiên.

2.5 Độ đo hiệu suất mô hình

2.5.1 Mô hình trích xuất đặc trưng

Vì đây là một mô hình phân loại nhiều nhãn, ta sẽ cần tập trung vào các chỉ số đánh giá hiệu quả đặc thù cho loại mô hình này, khác với phân loại đơn nhãn. Cụ thể:

- Micro: Tính toán các số liệu (precision, recall, F1) trên toàn bộ các dự đoán, coi tất cả nhãn như một tập hợp lớn. Ưu tiên khi quan tâm đến hiệu suất tổng thể.

- Macro: Tính toán các số liệu cho từng nhãn riêng lẻ, rồi lấy trung bình. Ưu tiên khi muốn đảm bảo mỗi nhãn đều được đánh giá công bằng, không bị các nhãn lớn lấn át.

2.5.2 Mô hình sinh nhạc

2.5.2.1 Độ đo ASA

2.5.2.2 Độ đo Rouge

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) là một bộ các phương pháp đo lường tự động được sử dụng để đánh giá chất lượng của văn bản được sinh ra bằng cách so sánh nó với một hoặc nhiều văn bản gốc. ROUGE được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên, đặc biệt là trong nhiệm vụ tóm tắt văn bản.

2.5.2.3 Độ đo dựa trên lý thuyết âm nhạc

2.5.2.4 Độ đo dựa trên lý thuyết âm nhạc

2.5.2.5 Độ đo dựa trên lý thuyết âm nhạc

Chương 3

Phương pháp đề xuất

Chương này mô tả chi tiết phương pháp thực hiện, các chiến lược đề xuất trong khoá luận.

3.1 Xác định chi tiết vấn đề

Đây là bài toán biến đổi ngôn từ tự nhiên thành âm nhạc, nghĩa là từ một câu đầu vào cụ thể, mô hình sẽ tạo ra một bản nhạc tương ứng. Thách thức của bài toán này là sản phẩm âm nhạc phải không chỉ phù hợp với nội dung và cảm xúc của câu văn đầu vào mà còn phải có tính thẩm mỹ, sáng tạo và hấp dẫn về mặt âm nhạc. Để đạt được điều này, việc lựa chọn kiến trúc mô hình phù hợp và sở hữu nguồn dữ liệu chất lượng cao là rất quan trọng. Phần bên dưới sẽ đi sâu về phần chi tiết về kiến trúc mô hình, phương pháp huấn luyện cũng như dữ liệu.

3.2 Tổng quan về phương pháp

3.2.1 Kiến trúc tổng quát

Kiến trúc sẽ đi qua hai giai đoạn:

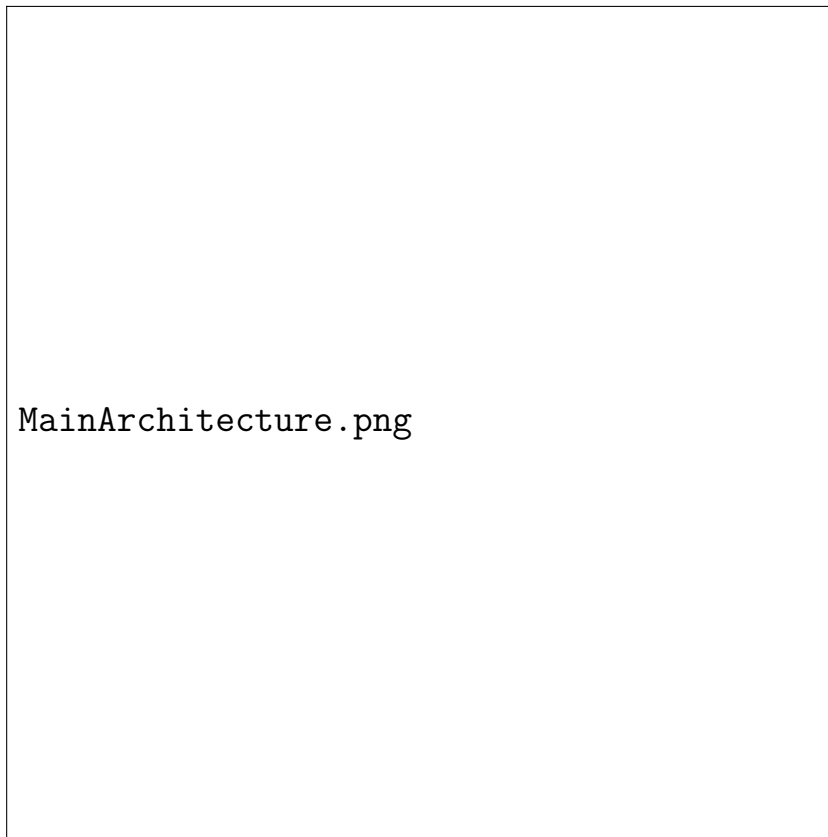
Giai đoạn 1 là trích xuất các thuộc tính âm nhạc dựa trên các nhãn đã

có sẵn, giai đoạn thứ hai là từ các nhãn sau khi đã được trích xuất đó thì đưa vào mô hình ngôn ngữ để dự đoán ra bài nhạc.

Giai đoạn 1: Trích xuất các thuộc tính âm nhạc. Giai đoạn này tập trung vào việc xác định các đặc trưng âm nhạc dựa trên sở thích và yêu cầu của nhạc sĩ, nhà sản xuất âm nhạc, hoặc người yêu thích âm nhạc. Chúng tôi thu thập thông tin về các yêu cầu như loại nhạc cụ được yêu thích (ví dụ, piano hoặc guitar), tốc độ của bản nhạc (nhanh hay chậm), và cao độ của giai điệu. Từ những thông tin này, chúng tôi tổng hợp và gán nhãn cho dữ liệu âm nhạc để phản ánh các thuộc tính này.

Giai đoạn 2: Sinh nhạc từ các nhãn đã trích xuất. Các thuộc tính âm nhạc đã được gán nhãn trong giai đoạn đầu được sử dụng để tạo ra câu dẫn (prompt) cho mô hình ngôn ngữ. Mô hình này sau đó sử dụng câu dẫn để sinh ra bản nhạc phù hợp với các đặc trưng đã được nêu. Quá trình này không chỉ nhằm đảm bảo bản nhạc tạo ra phải chính xác về mặt ngữ cảnh mà còn cần phải sáng tạo và hấp dẫn về mặt âm nhạc, phù hợp với yêu cầu thẩm mỹ.

Mỗi giai đoạn đều có vai trò quan trọng trong việc đạt được mục tiêu cuối cùng của dự án: tạo ra bản nhạc không chỉ thỏa mãn yêu cầu kỹ thuật mà còn đáp ứng được yếu tố nghệ thuật và cảm xúc.



Hình 3.1: Kiến trúc tổng quát của mô hình

3.2.2 Lí do lựa chọn

Trong nghiên cứu này, chúng tôi quyết định sử dụng kiến trúc hai lớp để giải quyết bài toán sinh nhạc từ ngôn ngữ tự nhiên. Câu hỏi đặt ra là tại sao chúng tôi lại chọn kiến trúc này thay vì các kiến trúc khác thường được sử dụng trong các mô hình ngôn ngữ.

Thông thường, các mô hình ngôn ngữ lớn thường sử dụng một trong hai loại kiến trúc: mô hình decoder như GPT để sinh văn bản, hoặc mô hình encoder như BERT để tóm tắt và trích xuất thông tin từ văn bản. Tuy nhiên, một nhược điểm lớn của các mô hình này là lượng tài nguyên tính toán và dữ liệu cần thiết để huấn luyện lại từ đầu là cực kỳ lớn. Ví dụ, việc huấn luyện từ đầu một mô hình ngôn ngữ lớn như GPT-3.5 hay Llama2 không chỉ tốn kém chi phí mà còn đòi hỏi lượng tài nguyên khổng lồ, điều này là không khả thi trong nhiều trường hợp cụ thể và đôi khi

không mang lại hiệu quả cao cho các tác vụ chuyên biệt.

Trong lĩnh vực âm nhạc, các thuộc tính phổ biến thường chỉ tập trung vào một lượng nhất định, và chúng tôi cũng chỉ tập trung vào những thuộc tính này. Các thông tin của một bản nhạc thường chỉ bao gồm một số lượng từ vựng nhất định, giúp giảm bớt độ phức tạp khi xử lý và huấn luyện mô hình.

Trong khi đó, các thuộc tính âm nhạc mà chúng tôi muốn mô hình hóa lại tập trung vào một lượng từ vựng nhất định và không cần đến toàn bộ bộ từ vựng rộng lớn như trong các mô hình GPT3.5 hay Llama2. Việc này cho phép chúng tôi sử dụng các mô hình nhỏ hơn và hiệu quả hơn như GPT-2 và Transformer casual language model, đồng thời tận dụng bộ từ điển REMI đã được tối ưu hóa cho âm nhạc, giảm số lượng từ vựng từ hơn 50,000 xuống còn 1,253 từ.

Ngoài ra, việc áp dụng kỹ thuật LoRA (Low-Rank Adaptation) cho phép chúng tôi tinh chỉnh mô hình với chỉ một phần nhỏ các tham số của nó, giảm bớt nhu cầu về tài nguyên trong khi vẫn duy trì hiệu suất cao. Điều này cũng hỗ trợ cho việc huấn luyện và triển khai mô hình trở nên linh hoạt và hiệu quả hơn.

Cuối cùng, sự độc lập của hai mô hình trong kiến trúc hai lớp này cũng làm cho việc xử lý dữ liệu trở nên đơn giản hơn, vì dữ liệu có thể được quản lý một cách hiệu quả hơn, tập trung vào các đặc tính kỹ thuật thay vì phải đối mặt với sự phức tạp của ngôn ngữ tự nhiên và các cách mô tả âm nhạc khác nhau. Việc này giúp cải thiện độ chính xác và giảm thiểu nhiễu trong quá trình huấn luyện và sinh sản phẩm cuối cùng. Từ bài toán sinh nhạc bằng ngôn ngữ tự nhiên ban đầu, chúng tôi đã chuyển đổi được thành hai bài toán nhỏ hơn.

Nhờ vào những lựa chọn này, dự án không chỉ đạt được mục tiêu ban đầu về sinh nhạc từ ngôn ngữ tự nhiên mà còn đảm bảo hiệu quả và khả thi về mặt tài nguyên, mở ra khả năng áp dụng rộng rãi.

3.2.3 Các bước thực hiện

Xử lý dữ liệu, trích xuất thông tin từ câu nhập vào, và sinh nhạc. Quy trình này trực quan qua sơ đồ hình 3.1, và được tóm tắt như sau:

1. Thu thập, tiền xử lý dữ liệu và trực quan hóa dữ liệu. Bao gồm việc thu thập dữ liệu thô, làm sạch và chuẩn hóa dữ liệu để phù hợp với nhu cầu của mô hình. Các bước này cũng bao gồm việc trực quan hóa dữ liệu để hiểu rõ hơn về cấu trúc và mối quan hệ trong dữ liệu.
2. Huấn luyện mô hình trích xuất đặc trưng âm nhạc: Phát triển và huấn luyện một mô hình để xác định và trích xuất các đặc trưng âm nhạc quan trọng từ dữ liệu đầu vào. Các đặc trưng này bao gồm nhân về nhạc cụ, tốc độ bài hát, tông cao, và các yếu tố khác.
3. Huấn luyện mô hình sinh âm nhạc: Sử dụng các đặc trưng đã trích xuất để huấn luyện một mô hình sinh âm nhạc có khả năng tạo ra các bản nhạc mới dựa trên các câu dẫn đã được định nghĩa.
4. Đánh giá hiệu suất của mô hình: Kiểm tra hiệu suất của mô hình dựa trên các tiêu chí đã được đề cập trong chương trước, như độ chính xác của các đặc trưng âm nhạc được sinh ra và tính thẩm mỹ của bản nhạc.
5. So sánh và đánh giá kết quả từ các mô hình: So sánh hiệu suất giữa các mô hình khác nhau và các phương pháp huấn luyện để chọn ra mô hình tối ưu nhất.
6. Chuyển đổi các đặc trưng thành câu dẫn: Biến các đặc trưng âm nhạc thành câu dẫn sử dụng bộ từ điển đã cho trước và đưa chúng vào mô hình sinh nhạc.
7. Hậu xử lý nếu có. Đảm bảo kết quả ra được bài nhạc.

3.3 Chuẩn bị dữ liệu

Chuẩn bị dữ liệu là bước vô cùng quan trọng trong việc xây dựng các mô hình trí tuệ nhân tạo. Trong phần này, chúng tôi tập trung vào việc thu thập và tiền xử lý dữ liệu ngôn ngữ tự nhiên và dữ liệu âm nhạc, với các mục tiêu cụ thể liên quan đến các bộ dữ liệu MuseCoco, MusicCaps, và Hooktheory (đã giới thiệu ở phần 2.3.1).

3.3.1 Dữ liệu ngôn ngữ tự nhiên

Mục tiêu đầu ra của công đoạn chuẩn bị dữ liệu ngôn ngữ tự nhiên là **các mẫu câu (template) mô tả âm nhạc** bằng tiếng Anh và tiếng Việt. Giá trị của các thuộc tính âm nhạc trong câu mô tả bị che bằng các nhãn (label) tương ứng. Dữ liệu này sẽ hỗ trợ trong quá trình huấn luyện Masked Language Model (kỹ thuật được giới thiệu ở mục 2.1.2.1).

3.3.1.1 Thu thập

Attributes	Values
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. Each instrument: 0: played, 1: not played, 2: NA
Pitch	Range: 0-11: octaves, 12: NA
Rhythm Danceability	0: danceable, 1: not danceable, 2: NA
Rhythm Intensity	0: serene, 1: moderate, 2: intense, 3: NA

Continued on next page

Bảng 3.1 – continued from previous page

Attributes	Values
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: other tempos, 7: NA
Key	0: major, 1: minor, 2: NA
Tempo	0: slow (≤ 76 BPM), 1: moderato (76-120 BPM), 2: fast (≥ 120 BPM), 3: NA
Time	0: 0-15s, 1: 15-30s, 2: 30-45s, 3: 45-60s, 4: >60 s, 5: NA
Artist	0-16 artists: Beethoven, Mozart, Chopin, Schubert, Schumann, J.S.Bach, Haydn, Brahms, Handel, Tchaikovsky, Mendelssohn, Dvorak, Liszt, Stravinsky, Mahler, Prokofiev, Shostakovich, 17: NA
Genre	21 genres: new age, electronic, rap, religious, international, easy listening, avant garde, RNB, latin, children, jazz, classical, comedy, pop, reggae, stage, folk, blues, vocal, holiday, country, symphony Each genre: 0: with, 1: without, 2: NA
Emotion	0-3: the 1-4 quartiles

3.3.1.2 Tiền xử lý

3.3.1.3 Khai phá dữ liệu

3.3.2 Dữ liệu âm nhạc

Mục tiêu của công đoạn chuẩn bị dữ liệu âm nhạc là tạo ra một bộ dữ liệu dưới định dạng MuseCoco (đã giới thiệu ở mục 2.3.3.4) từ các bản nhạc thiên về các bản nhạc hiện đại và có tính cập nhật so với các bộ dữ liệu trước đây vốn tập trung vào nhạc cổ điển, rock, và pop theo phong cách của nhiều thập niên trước.

3.3.2.1 Thu thập

3.3.2.2 Tiền xử lý

3.3.2.3 Khai phá dữ liệu

3.4 Huấn luyện mô hình

3.4.1 Mô hình trích xuất đặc trưng câu văn bản

Dự án này ứng dụng mô hình BERT, vốn là một mô hình thuộc loại encoder có khả năng trích xuất đặc trưng văn bản hiệu quả, để dự đoán các thuộc tính âm nhạc từ câu văn nhập vào. Dựa vào cấu trúc của mô hình BERT, nhóm nghiên cứu đã phát triển "MusicBert", một biến thể được tinh chỉnh để phù hợp với nhiệm vụ phân loại các thuộc tính âm nhạc đa dạng. Nhóm chúng tôi đã áp dụng BERT cho việc dự đoán các thuộc tính xuất hiện trong câu như, có tiếng đàn piano hay không, thời thượng bản nhạc dài bao lâu, tốc độ nhanh khoảng bao nhiêu, có được đề cập hay không, kết quả sau khi được dự đoán sẽ đi qua một hàm Softmax để phân loại vào các nhánh cho trước. Gồm có hai loại thuộc tính là thuộc tính định tính và định lượng. Thuộc tính nhạc cụ (định tính) sẽ được phân loại theo 3 nhãn gồm: có xuất hiện, không xuất hiện, không được đề cập. Thuộc tính định lượng đo về đại lượng sẽ được sắp xếp theo từng mức độ đã quy định sẵn ví dụ như thời lượng thì sẽ có từ 0-15s hay 15-30s.

Từ kiến trúc BERT gốc ban đầu chúng tôi đã lên ý tưởng, tham khảo từ nhiều nguồn và đã cài đặt để phù hợp cho các tác vụ phân loại nhiều lớp như sau:

- Token [CLS]: Được thêm vào trước khi huấn luyện, giúp phân loại với 53 nhãn tương ứng cho các thuộc tính nhạc cụ, tốc độ bài hát, và thời lượng.
- Lớp Pooler: Chịu trách nhiệm lấy ra các vector đại diện cho token [CLS] từ Embeddings.

- Hàm mất mát: Sử dụng CrossEntropyLoss cho việc phân loại đa nhãn.
- Lớp Softmax: Được sử dụng ở cuối để xác định xác suất của mỗi nhãn.
- Những lớp còn lại giống với mô hình gốc.

Mô hình trích xuất đặc trưng âm nhạc được tinh chỉnh và huấn luyện từ bộ trọng số của mô hình gốc là bert-large-multilingual-uncased, mô hình này hỗ trợ đa ngôn ngữ. Do đó, chúng tôi đã sử dụng tập dữ liệu tiếng Anh, tiếng Việt để huấn luyện.

Trước khi đưa dữ liệu vào thì cần phải qua bước tiền xử lý huấn luyện. Sử dụng tokenizer để mã hóa văn bản đầu vào đi kèm theo các tham số như padding, max-length, truncation. Nhãn là một chuỗi one-hot tương ứng., với giá trị nào được đánh dấu là 1 thì được phân loại vào lớp đó. Kết quả sẽ trả ra đầu vào text đã được mã hóa để mô hình có thể hiểu được.

Về hàm datacollator hay gom dữ liệu, là thao tác để tổ chức dữ liệu huấn luyện trước khi đưa vào mô hình. Hàm này nhận danh sách các đặc trưng của dữ liệu đầu vào và trả về một từ điển chứa các tensors để sử dụng trong quá trình huấn luyện. Bước này, chúng tôi đã xác định dữ liệu nhãn và tạo tensors cho chúng. Nếu nhãn không phải là None và là một tensor, nó sẽ được chuyển đổi sang kiểu dữ liệu phù hợp (long hoặc float) và chuyển thành tensor.

Tiếp theo ta đi sâu vào quá trình huấn luyện MusicBert. Ở đây chúng tôi đã sử dụng CrossEntropyLoss để tính toán hàm mất mát, đồng thời đánh giá được hiệu suất và độ chính xác của mô hình đưa ra. Bên cạnh đó thì cũng đã sử dụng thuật toán tối ưu hóa và cập nhật trọng số Adam với hệ số beta1 trong thuật toán Adam, được sử dụng để tính toán mức độ thay đổi trung bình của gradient là 0.9 và beta2 là 0.999. Learning rate ban đầu được đặt là 0.0001. Mô hình được huấn luyện trên 100 epochs để cho ra kết quả tốt nhất.

Với việc huấn luyện mô hình đa ngôn chúng tôi đã lựa chọn cách thức huấn luyện các mô hình riêng biệt cho từng ngôn ngữ, sử dụng dữ liệu huấn luyện chỉ thuộc ngôn ngữ đó. Do mô hình có thể tập trung vào việc học các đặc trưng riêng của từng ngôn ngữ, có thể dẫn đến hiệu suất tốt hơn trên từng ngôn ngữ cụ thể (tiếng Anh, tiếng Việt).

Bảng bên dưới thể hiện chi tiết từng tham số mà chúng tôi đã lựa chọn.

Tham số	Giá trị	Ý nghĩa
epoch	5	Số lượng epoch xác định số lần mô hình sẽ được huấn luyện trên toàn bộ dữ liệu.
batch size	32	Kích thước của mỗi batch dữ liệu đưa vào mô hình trong mỗi bước huấn luyện.
optimizer	AdamW	Bộ tối ưu hóa sử dụng để cập nhật các tham số của mô hình trong quá trình huấn luyện.
learning rate	2e-5	Tốc độ học xác định mức độ điều chỉnh các trọng số của mô hình sau mỗi bước huấn luyện.
max sequence length	128	Chiều dài tối đa của chuỗi đầu vào, các chuỗi dài hơn sẽ bị cắt ngắn.
warmup steps	500	Số bước đầu tiên trong đó tốc độ học tăng dần đến giá trị tối đa đã định.
dropout rate	0.1	Tỷ lệ dropout sử dụng trong quá trình huấn luyện để tránh overfitting.
gradient accumulation steps	2	Số bước gradient accumulation trước khi cập nhật trọng số mô hình.
weight decay	0.01	Hệ số điều chỉnh tỷ lệ suy giảm trọng số, giúp tránh overfitting.

Bảng 3.2: Các hyperparameter được sử dụng trong quá trình huấn luyện mô hình BERT với lớp Softmax

3.4.2 Mô hình sinh nhạc

Khi bắt đầu với đề tài nghiên cứu về mô hình ngôn ngữ, một trong những thách thức lớn nhất là lựa chọn kiến trúc và mô hình sao cho phù hợp với nguồn tài nguyên sẵn có nhưng vẫn đảm bảo cho ra kết quả tốt. Điều này đặt ra một bài toán cân bằng giữa hiệu suất của mô hình và tài nguyên tính toán. Nếu mô hình ngôn ngữ gốc chưa 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 bối cảnh này, sự ra đời của Low-Rank Adaptation (LoRA) đã mang lại một giải pháp hiệu quả để giải quyết vấn đề này. Bài luận này sẽ thảo luận về các yếu tố cần xem xét khi lựa chọn mô hình ngôn ngữ, vai trò của LoRA, và cách LoRA có thể giúp tối ưu hóa tài nguyên và hiệu suất.

Trong quá trình huấn luyện, nhóm nghiên cứu của chúng tôi đã tiến

hành thử nghiệm với hai phương pháp khác nhau để xác định phương pháp tối ưu nhất cho mục tiêu của dự án. Hai mô hình chính được sử dụng bao gồm:

- **GPT-2 LoRA:** Sử dụng kiến trúc GPT-2 đã được tối ưu bằng cách áp dụng phương pháp Low-Rank Adaptation (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.
- **Transformer Seq2Seq Sử Dụng Fairseq:** Là một phương pháp tiếp cận khác sử dụng kiến trúc Transformer dạng sequence-to-sequence, được triển khai qua bộ công cụ Fairseq của Facebook AI Research. Kiến trúc Seq2Seq này được thiết kế để xử lý các nhiệm vụ biến đổi chuỗi, như dịch máy hoặc tổng hợp văn bản, và được tối ưu hóa để cung cấp một giải pháp hiệu quả cho việc sinh nhạc dựa trên ngôn ngữ tự nhiên.

3.4.2.1 GPT-2 LoRA

3.4.2.2 Transformer Seq2Seq Sử Dụng Fairseq

Để đánh giá hiệu quả của hai mô hình, chúng tôi đã thiết lập một loạt các thí nghiệm, trong đó mỗi mô hình được huấn luyện và đánh giá dựa trên cùng một tập dữ liệu. Các bài kiểm tra được thiết kế để đánh giá khả năng của mỗi mô hình trong việc xử lý và tạo ra âm nhạc phù hợp với các yêu cầu đặt ra, từ đó so sánh hiệu suất của chúng dựa trên các tiêu chuẩn như độ chính xác, tốc độ xử lý và hiệu quả sử dụng tài nguyên. Linear Casual Language Model là một mô hình ngôn ngữ đơn giản, sử dụng tuyến tính và dựa trên sự tuần tự của các từ trong một câu để dự đoán từ tiếp theo. Mô hình này có những ưu điểm như dễ triển khai, yêu cầu tài nguyên thấp, và thời gian huấn luyện nhanh hơn so với các mô hình phức tạp. Tuy nhiên, LCM cũng có những hạn chế như khả năng bắt kịp ngữ cảnh dài

hạn không tốt bằng các mô hình như Transformer.

Kết quả cho thấy mô hình Linear Casual Language Model có thể đạt được độ chính xác tương đối cao trong các tác vụ tạo văn bản ngắn và trung bình. Mặc dù LCM không thể bắt kịp ngữ cảnh dài hạn tốt như các mô hình Transformer Seq2Seq, nhưng nó vẫn tạo ra các văn bản có độ trôi chảy và phù hợp ngữ cảnh trong phạm vi ngắn.

Tham số	Giá trị	Ý nghĩa
epoch	100	Số lượng epoch xác định số lần mô hình sẽ được huấn luyện trên toàn bộ dữ liệu.

Bảng 3.3:

3.4.2.3 LoRA GPT

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, huấn luyện và đánh giá kết quả. Dưới đây là mô tả chi tiết từng bước và bảng các hyperparameter sử dụng trong quá trình huấn luyện.

Dữ liệu được token hóa bằng cách sử dụng tokenizer với các tham số như độ dài tối đa và padding. Hàm tokenize sẽ xử lý dữ liệu đầu vào để phù hợp với mô hình.

Mô hình GPT-2 được cấu hình với các tham số như số lớp transformer, số lượng head trong multi-head attention, kích thước embedding, và các token đặc biệt. Các tham số này đảm bảo rằng mô hình có đủ khả năng để học và dự đoán chính xác.

Kỹ thuật LoRA được cấu hình để cải thiện hiệu suất của mô hình. Chúng tôi thiết lập các tham số như r , lora_α , $\text{lora}_\text{dropout}$, vccmodule và Och.Kthutn .

Chúng tôi cấu hình các tham số huấn luyện như số epoch, batch size, optimizer, learning rate, và các chiến lược lưu và đánh giá mô hình. Sử dụng lớp Trainer của transformers, mô hình được huấn luyện với các dữ liệu đã token hóa và cấu hình huấn luyện đã thiết lập.

Tham số	Giá trị	Ý nghĩa
epoch	10	Số lượng epoch xác định số lần mô hình sẽ được huấn luyện trên toàn bộ dữ liệu.
batch size	1	Kích thước của mỗi batch dữ liệu đưa vào mô hình trong mỗi bước huấn luyện.
optimizer	AdamW	Bộ tối ưu hóa sử dụng để cập nhật các tham số của mô hình trong quá trình huấn luyện.
learning rate	5e-4	Tốc độ học xác định mức độ điều chỉnh các trọng số của mô hình sau mỗi bước huấn luyện.
max sequence length	2048	Chiều dài tối đa của chuỗi đầu vào, các chuỗi dài hơn sẽ bị cắt ngắn.
warmup steps	2000	Số bước đầu tiên trong đó tốc độ học tăng dần đến giá trị tối đa đã định.
dropout rate	0.1	Tỷ lệ dropout sử dụng trong quá trình huấn luyện để tránh overfitting.
gradient accumulation steps	2	Số bước gradient accumulation trước khi cập nhật trọng số mô hình.
weight decay	0.01	Hệ số điều chỉnh tỷ lệ suy giảm trọng số, giúp tránh overfitting.
lora r	16	Hệ số r trong kỹ thuật LoRA xác định số chiều của ma trận low-rank.
lora alpha	12	Tham số alpha trong kỹ thuật LoRA, điều chỉnh mức độ ảnh hưởng của ma trận low-rank.
lora dropout	0.1	Tỷ lệ dropout áp dụng trong kỹ thuật LoRA để tránh overfitting.

Bảng 3.4: Các hyperparameter được sử dụng trong quá trình huấn luyện mô hình GPT-2 với lớp Softmax và kỹ thuật LoRA

3.5 Hậu xử lý dữ liệu

3.6 Chọn phương pháp đánh giá mô hình

3.6.1 Các thông số đánh giá của nghiên cứu có trước

3.6.1.1 Đánh giá mô hình trích xuất đặc trưng

Tham số	Giá trị	Ý nghĩa
epoch	10	Số lượng epoch xác định số lần mô hình sẽ được huấn luyện trên toàn bộ dữ liệu.
batch size	1	Kích thước của mỗi batch dữ liệu đưa vào mô hình trong mỗi bước huấn luyện.
optimizer	AdamW	Bộ tối ưu hóa sử dụng để cập nhật các tham số của mô hình trong quá trình huấn luyện.
learning rate	5e-4	Tốc độ học xác định mức độ điều chỉnh các trọng số của mô hình sau mỗi bước huấn luyện.
max sequence length	2048	Chiều dài tối đa của chuỗi đầu vào, các chuỗi dài hơn sẽ bị cắt ngắn.
warmup steps	2000	Số bước đầu tiên trong đó tốc độ học tăng dần đến giá trị tối đa đã định.
dropout rate	0.1	Tỷ lệ dropout sử dụng trong quá trình huấn luyện để tránh overfitting.
gradient accumulation steps	2	Số bước gradient accumulation trước khi cập nhật trọng số mô hình.
weight decay	0.01	Hệ số điều chỉnh tỷ lệ suy giảm trọng số, giúp tránh overfitting.
lora r	16	Hệ số r trong kỹ thuật LoRA xác định số chiều của ma trận low-rank.
lora alpha	12	Tham số alpha trong kỹ thuật LoRA, điều chỉnh mức độ ảnh hưởng của ma trận low-rank.
lora dropout	0.1	Tỷ lệ dropout áp dụng trong kỹ thuật LoRA để tránh overfitting.

Bảng 3.5: Các hyperparameter được sử dụng trong quá trình huấn luyện mô hình GPT-2 với lớp Softmax và kỹ thuật LoRA

3.6.2 Các thông số đánh giá nhóm bổ sung

Chương 4

Kết quả thí nghiệm

4.1 Kết quả huấn luyện

4.1.1 Mô hình tích xuất đặc trưng

Trong bài luận này, chúng tôi sẽ trình bày kết quả huấn luyện của mô hình GPT-2 với lớp Softmax và kỹ thuật LoRA (Low-Rank Adaptation). Mục tiêu là đánh giá hiệu suất của mô hình trong việc dự đoán giá trị tương lai dựa trên dữ liệu đầu vào. Các chỉ số đánh giá chính bao gồm MAE (Mean Absolute Error), RMSE (Root Mean Square Error) và MAPE (Mean Absolute Percentage Error).

Attribute	Accuracy (%)	Attribute	Accuracy (%)
I_piano	100.00	I_clarinet	99.92
I_keyboard	99.92	I_piccolo	99.94
I_percussion	100.00	I_flute	99.62
I_organ	100.00	I_pipe	100.00
I_guitar	99.92	I_synthesizer	100.00
I_bass	99.84	I_ethnic_instruments	99.98
I_violin	99.92	I_sound_effects	99.98
I_viola	99.96	I_drum	100.00
I_cello	99.92	Genre_new_age	99.98
I_harp	100.00	Genre_electronic	100.00
I_strings	99.96	Genre_rap	100.00
I_voice	99.70	Genre_religious	100.00
I_trumpet	99.96	Genre_international	100.00
I_trombone	99.94	Genre_easy_listening	100.00
I_tuba	100.00	Genre_avant_garde	100.00
I_horn	99.94	Genre_rnb	100.00
I_brass	100.00	Genre_latin	100.00
I_sax	99.84	Genre_children	100.00
I_oboe	99.94	Genre_jazz	100.00
I_bassoon	99.96	Genre_classical	100.00
Genre_comedy_spoken	100.00	Genre_pop_rock	100.00
Genre_reggae	100.00	Genre_stage	100.00
Genre_folk	100.00	Genre_blues	100.00
Genre_vocal	100.00	Genre_holiday	100.00
Genre_country	100.00	Genre_symphony	100.00
Bar	100.00	Time Signature	100.00
Key	100.00	Tempo	99.84
Octave	100.00	Emotion	99.80
Time	100.00	Rhythm Danceability	100.00
Rhythm Intensity	99.88	Artist	100.00

Bảng 4.1: Độ chính xác của các thuộc tính âm nhạc

Bảng trên liệt kê độ chính xác của các thuộc tính âm nhạc sau quá trình huấn luyện mô hình. Kết quả được đo lường bằng phần trăm (

4.1.2 Mô hình tích xuất đặc trưng

Trong bài luận này, chúng tôi sẽ trình bày kết quả huấn luyện của mô hình GPT-2 với lớp Softmax và kỹ thuật LoRA (Low-Rank Adaptation).

Mục tiêu là đánh giá hiệu suất của mô hình trong việc dự đoán giá trị tương lai dựa trên dữ liệu đầu vào. Các chỉ số đánh giá chính bao gồm MAE (Mean Absolute Error), RMSE (Root Mean Square Error) và MAPE (Mean Absolute Percentage Error).

Độ chính xác cao: Các thuộc tính nhạc cụ như I_{piano} , $I_{percussion}$, I_{organ} , I_{harp} ,

Hiệu suất xuất sắc: Kết quả cho thấy mô hình đạt hiệu suất xuất sắc trong việc nhận diện và phân loại các thuộc tính âm nhạc. Độ chính xác gần như tuyệt đối trên hầu hết các thuộc tính chứng tỏ mô hình đã được huấn luyện tốt và có khả năng tổng quát hóa cao. Độ tin cậy: Với độ chính xác cao như vậy, mô hình có thể được áp dụng vào thực tế để hỗ trợ các tác vụ liên quan đến âm nhạc như nhận diện nhạc cụ, phân loại thể loại nhạc, và phân tích các thuộc tính âm nhạc khác. Cải thiện nhỏ: Mặc dù kết quả rất tốt, vẫn có không gian để cải thiện cho một số thuộc tính có độ chính xác dưới 100

4.1.3 Mô hình tích xuất đặc trưng

TODO	TODO	TODO	TODO
TODO	TODO	TODO	TODO

Bảng 4.2:

4.2 Phần mềm demo

Ứng dụng sinh nhạc bằng văn bản sử dụng kiến trúc client-server là một hệ thống cho phép người dùng nhập vào văn bản để tạo ra nhạc dựa trên nội dung văn bản đó. Hệ thống này sử dụng mô hình học sâu để

phân tích và chuyển đổi văn bản thành nhạc. Kiến trúc client-server giúp tách biệt các thành phần xử lý, đảm bảo hiệu suất và dễ dàng mở rộng hệ thống.

Kiến Trúc Tổng Quan Kiến trúc client-server của ứng dụng sinh nhạc bao gồm hai thành phần chính:

Client: Giao diện người dùng, nơi người dùng nhập văn bản và nghe nhạc đã được tạo ra. Server: Xử lý các yêu cầu từ client, bao gồm phân tích văn bản và sinh nhạc dựa trên văn bản đó.

4.2.0.1 Mô Tả Chi Tiết

1. Client Client là ứng dụng phía người dùng, có thể là một ứng dụng web, ứng dụng di động hoặc ứng dụng desktop. Các chức năng chính của client bao gồm:

Giao diện người dùng: Cung cấp giao diện để người dùng nhập văn bản và tương tác với hệ thống. Gửi yêu cầu: Gửi yêu cầu chứa văn bản cần chuyển đổi đến server. Nhận phản hồi: Nhận nhạc đã được sinh từ server và phát nhạc cho người dùng nghe. Luồng hoạt động của client:

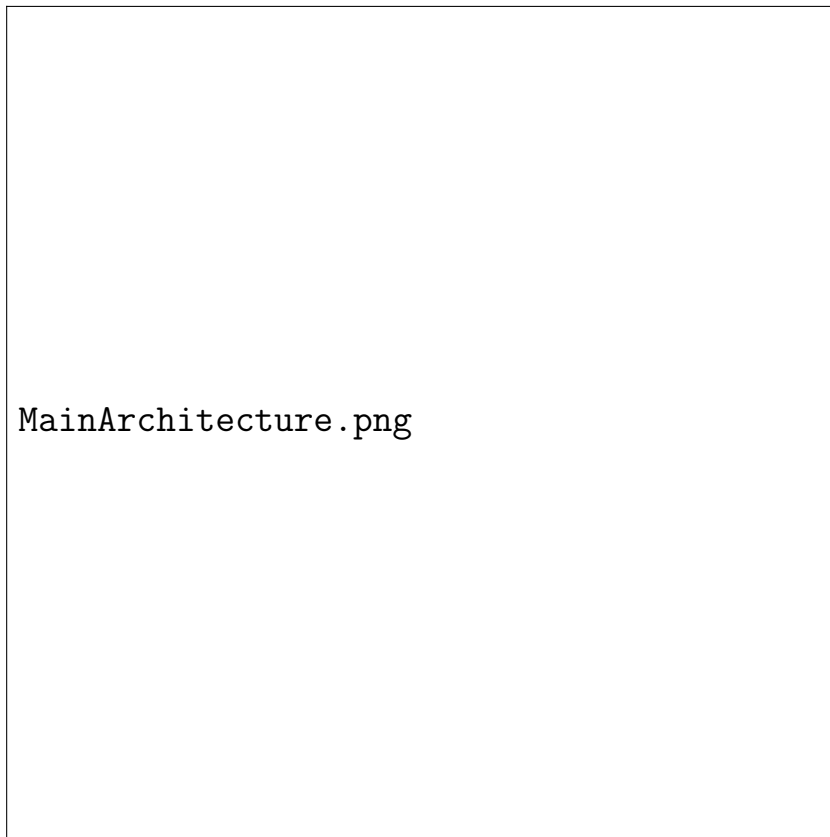
Người dùng nhập văn bản vào giao diện. Client gửi yêu cầu chứa văn bản đến server. Client nhận nhạc đã được sinh từ server. Client phát nhạc cho người dùng.



Hình 4.1: Kiến trúc tổng quát của mô hình

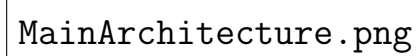
2. Server Server là thành phần xử lý chính của hệ thống, bao gồm các chức năng sau:

API xử lý yêu cầu: Tiếp nhận và xử lý các yêu cầu từ client. Mô hình học sâu: Phân tích văn bản và sinh nhạc dựa trên mô hình học sâu đã được huấn luyện. Cơ sở dữ liệu: Lưu trữ các bản nhạc đã sinh và thông tin người dùng nếu cần thiết. Luồng hoạt động của server:



Hình 4.2: Kiến trúc tổng quát của mô hình

Server nhận yêu cầu chứa văn bản từ client. Server phân tích văn bản bằng mô hình học sâu. Server sinh nhạc dựa trên nội dung văn bản. Server gửi nhạc đã sinh trở lại client.

The image is a large, empty rectangular box with a thin black border. Inside the box, the text "MainArchitecture.png" is written in a monospaced font, positioned towards the left side. This likely represents a missing or placeholder image for a system architecture diagram.

MainArchitecture.png

Hình 4.3: Kiến trúc tổng quát của mô hình

Chương 5

Kết luận

Thông qua bài nghiên cứu của khóa luận này, chúng tôi đã tìm ra cách thay vì sử dụng một mô hình như truyền thống thì sử dụng hai mô hình nhỏ gọn hơn để xử lý tác vụ sinh nhạc từ câu ngôn ngữ tiếng Anh - tiếng Việt một cách có hiệu quả. Chúng tôi đã thử nghiệm trên nhiều kiến trúc mô hình hay hiệu chỉnh nhiều tham số nhằm đưa ra cái nhìn từ tổng quát đến chi tiết về khả năng sinh nhạc của kiến trúc trên cũng như việc so sánh các mô hình Transformer để kết luận được tác vụ phù hợp với mô hình nào hơn.

Các kỹ thuật về giảm việc ma trận LoRA giúp việc huấn luyện trở nên dễ dàng hơn, việc tinh chỉnh LoRA cũng được thử nghiệm qua các mô-đun nào lên chọn nhằm tối ưu việc huấn luyện cũng như cho ra kết quả tốt, thực nghiệm cho thấy rằng việc tinh chỉnh trên các lớp tuyến tính là điều cần thiết để đạt được hiệu suất tương đương với việc tinh chỉnh toàn bộ mô hình (các lớp bao gồm `gate_proj`, `down_proj`, `up_proj`, `q_proj`, `v_proj`, `k_proj`, `o_proj`) đóng vai trò quan trọng trong việc biến đổi và kết hợp thông tin. Ngoài ra, còn thực nghiệm tinh chỉnh trên nhiều bộ tham số đã đề cập ở trên nhằm cho ra kết quả tốt nhất.

Việc sử dụng các kỹ thuật casual language modelling trong các mô hình ngôn ngữ giúp phù hợp với tác vụ sinh nhạc

Việc đảm bảo cho kết quả luôn cho ra được bản nhạc còn cần thêm các kỹ thuật kiểm tra tính đúng đắn trong bài hát theo các nguyên tắc nhạc

lý, việc làm này được diễn ra sau khi mô hình sinh ra nhạc bằng quá trình vừa kiểm tra vừa điền chỗ còn thiếu bằng phương pháp nội suy cho ra kết quả tốt để ứng dụng được trong sản phẩm.

Mặc dù các phương pháp trên đều cho ra kết quả tương đối tốt, nhưng vẫn cần nghiên cứu thêm để điều chỉnh bộ các bộ tham số trên các tập dữ liệu phong phú và đa dạng hơn để đạt được mô hình đáp ứng cho nhu cầu về nhiều thể loại, màu sắc trong âm nhạc. Các thí nghiệm trên chỉ thực hiện trong quá trình huấn luyện bằng LoRA cho ra kết quả tốt cho nên chúng tôi nghĩ rằng nếu được huấn luyện với model gốc và đầy đủ sẽ cho ra kết quả tốt hơn.

Danh mục công trình của tác giả

1. Tạp chí ABC
2. Tạp chí XYZ

Tài liệu tham khảo

Tiếng Anh

- [4] Agostinelli, Andrea et al. *MusicLM: Generating Music From Text*. 2023. arXiv: 2301.11325 [cs.SD].
- [5] Hayes, Thomas et al. *MUGEN: A Playground for Video-Audio-Text Multimodal Understanding and GENeration*. 2022. arXiv: 2204.08058 [cs.CV].
- [6] Huang, Yu-Siang and Yang, Yi-Hsuan. *Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions*. 2020. arXiv: 2002.00212 [cs.SD].
- [7] Lu, Peiling et al. *MuseCoco: Generating Symbolic Music from Text*. 2023. arXiv: 2306.00110 [cs.SD].

Phụ lục A

Ngữ pháp tiếng Việt

Đây là phụ lục.

Phụ lục B

Ngữ pháp tiếng Nôm

Đây là phụ lục 2.