STYLE-VT: Variational Transformer와 대리 코드 이론을 적용한 스타일 기반 코드 생성 모델

송교원 ○ 서동영 나준수 양희철
충남대학교 컴퓨터융합학부
tim000519@gmail.com, lazymoon180@gmail.com, najunsoo4439@gmail.com hcyang@cnu.ac.kr

STYLE-VT: Style Conditioned Chord Generation By Variational Transformer With Chord Substitution Theory

Kyowon Song○ Dongyong Seo Junsoo Na Heecheol Yang
Division of Computer Convergence, Chungnam National University

요 약

음악 생성 분야에서 Transformer 모델은 음악의 코드 진행의 생성과 같은 작업에서 좋은 성능을 보여주고 있다. 하지만, 장르별 특성을 반영하여 코드 진행을 제어하는 연구는 진행되지 않았다. 본 연구에서는 Transformer 아키텍처와 변분 오토인코더(VAE)를 결합한 스타일 기반 코드 생성 모델인 Style-VT를 제안한다. 이 모델은 임베딩된 장르 정보를 활용하여 특정 장르의 스타일적 규범에 맞는 코드 진행을 생성한다. 또한 대리 코드 이론을 도입하여 목적 함수에 적용하고, 이를 통해 모델이 생성하는 코드 진행의 유연성을 높였다. 실험을 통해 Style-VT가 기존 방법들보다 더 유연한 코드 진행을 생성하고, 인간이 작곡한 음악과 더 높은 화성적 유사성을 달성함을 보였다.

1. 서 론

멜로디 기반 코드 생성(Melody conditioned chord generation)이란, 주어진 멜로디에 대해 어울리는 화음, 즉 코드의 진행을 붙여 화음을 이루는 음악을 생성하는 작업을 통칭한다. 멜로디 기반 코드 생성 연구의 초기 단계에서는 은닉 마르코프 모델[1]을 사용한 연구가 주로 진행되었고, 이 후 LSTM 기반의 딥러닝 모델이 도입되면서 성능이 비약적으로 향상되었다[2]. 그러나 위 접근 방식들은 구조화된 코드 진행을 생성하지 못한다는 한계를 가진다[3]. 이러한 한계는 어텐션 메커니즘을 이용한 Transformer 아키텍처가 도입되면서 효과적으로 해결되었고, 최근에는 Transformer 프레임워크에 다양한 기술을 통합하는 방식의 연구가 활발하게 진행되고 있다[3,4].

본 연구의 주 목적은 코드의 생성과정에 음악의 장르적 특성을 반영하는 것이다. 코드 진행은 음악에서 화성의 틀을 구성할 뿐만 아니라, 음악의 장르별 특징 또한 드러낼 수 있는 중요한 요소이다. 선행 연구 중 코드 진행을 기반으로 음악 장르를 분류하는 연구[5] 또는 멜로디를 포함한 전체 음악에 특정 스타일을 반영하는 연구[6]는 시도되었지만, 장르별 특성을 적용하여 생성되는 코드 진행을 제어하고자 하는 시도는 없었다. 본 연구는 Variational Transformer 아키텍처에 장르 임베딩을 통합한 스타일 기반 코드 생성 모델인 Style-VT를 제안한다. 장르 임베딩의 적용과 더불어, 음악 이론 중 하나인 대리 코드 이론을 도입하여 모델의 목적 함수를 재정의한다. 음악은 본질적으로 주관적인 특성이 강하여 동일한 멜로디에 대해서라도 다양한 코드 진행이 가능하고, 이러한 코드의 변형은 다른 장르로 음악을 편곡하는 데 필수적이다. 음악 이론[7]에 따르면, 코드의 대체(Chord substitution)는 코드 진행 내의 코드를 유사한 화성 기능을 수행하는 다른 코드로 대체하는 것을 의미한다. 우리는 이 대리 코드 체계를 공식화하고 이를 모델의 목적 함수에 적용함으로써 장르에 어울리는 코드 생성을 더욱 강화한다. 이 접근법은 기존 모델들이 원곡의 코드 진행만을 유일한 정답으로 취급하여 발생했던 한계를 해결하고, 더 유연하고 이론적인 코드 생성 접근법을 제시한다.

2. 제안 방법

2.1 대리 코드 이론

대리 코드 이론을 도입하기 위해서는 대리 코드에 대한 명확한 정의가 필요하지만, 기존 문헌[7]에서는 대리 코드의 범위에 대해 작곡자의 주관성이 허용되었다. 따라서 본 연구에서는 대리 코드의 범위를 수식으로써 명확히 정의하여 적용한다. 임의의 코드 C에 대해서, $C = \{\mathrm{p}_1, \mathrm{p}_2, \dots, \mathrm{p}_\mathrm{n}\}$라 하자. 여기서 p_i는 $i \in \{1, \dots, n\}$에 대해 n개의 음으로 구성된 코드 C의 음을 나타낸다. 임의의 두 코드 $C_a = \{p_{a1}, \dots, p_{an}\}$와 $C_b = \{p_{b1}, \dots, p_{bm}\}$에 대해 대리 코드는 다음과 같이 정의한다.

$$|C_a \cap C_b| \geq 3 \rightarrow C_b \in \mathcal{S}(C_a) \text{ and } C_a \in \mathcal{S}(C_b).$$

여기서 $\mathcal{S}(\mathrm{C_a})$와 $\mathcal{S}(\mathrm{C_b})$는 각각 $\mathrm{C_a}$와 $\mathrm{C_b}$의 대리 코드 집합을 나타낸다. 서로 다른 두 코드가 세 개 이상의 공통 구성음을 가질 경우, 각 코드는 서로에 대한 대리코드 집합에 속한다.

2.2 모델 설계

Style-VT는 Transformer와 변분 오토인코더 (VAE)로 구성된다. Transformer는 주어진 멜로디를 코드 시퀀스로 변환하는 역할을 하며, VAE는 장르에 대한 잠재 표현을 학습한다. 전체 모델 구조는 그림 1로 표현하였다.

입력 멜로디는 원 핫 벡터 시퀀스 $m_{1:T} \in \{0,1\}^{T \times |P|}$ 로 정의된다. 여기서 T는 멜로디의 길이, $|P|$는 음의 개수, t는

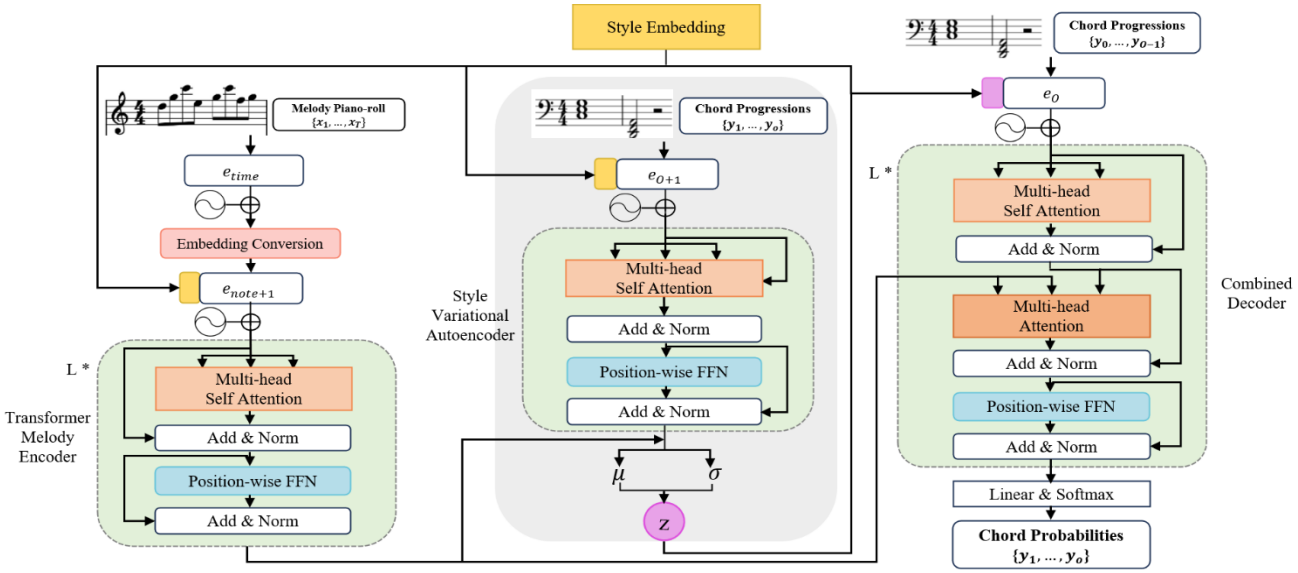

그림 1. 모델 구조

16분음표 단위의 시간 인덱스를 의미한다. 스타일 토큰 c는 장르 벡터와 조성 벡터가 결합된 벡터로, 스타일 정보를 각 모델 파트에 전달한다. Transformer Melody Encoder (TME)는 $m_{1:T}$를 입력으로 받은 후, 아래의 과정을 거쳐 멜로디에 대한 컨텍스트를 포착한다.

$$e_{time} = \text{Embedding}(m_{1:T}),$$
$$e_{note} = \text{EmbConv}(e_{time} + w_{time}, M),$$
$$e_{note+1} = \text{Concat}(c, e_{note}),$$
$$e'_{note+1} = \text{TMEs}(e_{note+1} + w_{note}).$$

Embedding과 TMEs는 각각 임베딩 층과 L개의 TME 블록을 의미한다. e_{time}, e_{note}, N은 각각 시간 축 임베딩 벡터, 음표 축 임베딩 벡터, 멜로디 음표의 개수를 나타낸다. w는 사인 함수 기반의 위치 임베딩을 나타내며, EmbConv는 멜로디 내의 음표 패턴을 포착하기 위해 시간 축 임베딩 e_{time}을 음표 축의 임베딩 e_{note}으로 변환하는 함수이다. 스타일 토큰 c는 e_{note}의 시작 부분에 결합된다. EmbConv 블록에서는 e_{time}에 위치 임베딩 w_{time}이 더해진 후, 정렬 매트릭스 $M \in \{0,1\}^{T \times N}$을 이용한 평균 풀링을 통해 음표 축 임베딩 e_{note}로 변환된다.

$$\text{EmbConv}(e, M) = \text{Linear}\left(\frac{M^T * e}{\sum_{t=1}^{T} M_{t,1:N}}\right).$$

Style Variational Encoder(SVE)는 Transformer 멜로디 인코더의 출력, 코드 입력 y, 그리고 스타일 토큰 c로부터 잠재 표현 z를 다음과 같은 과정을 통해 추론한다.

$$e_{O+1} = \text{Concat}(c, \text{Embedding}(y_{1:O})),$$
$$e'_{O+1} = \text{SVE}(e_{O+1} + w_{O+1}),$$
$$r = \text{Concat}(\text{Pool}(e'_{note+1}), \text{Pool}(e'_{O+1})),$$
$$[\mu, \sigma] = \text{Linear}(r), \quad \text{where } z \sim \mathcal{N}(\mu, \sigma).$$

$Pool$은 평균 풀링을 의미한다. Style Variational Encoder (SVE)와 Transformer Melody Encoder (TME)의 출력은 평균 집계되어 결합된 후 완전 연결 계층을 거쳐 두 개의 파라미터 μ와 σ를 생성한다. μ와 σ로부터 잠재 변수 z가 추론되며, 이 때 사전 분포는 정규 분포를 따른다고 가정한다. Combined Decoder(CD)은 스타일 토큰 c와 잠재 변수 z를 입력으로 다음 순서의 코드와 인코더 출력을 재구성한다. 그 과정은 다음과 같이 이루어진다.

$$e_O = \text{Concat}(z + c, Embedding(y_{1:O-1})),$$
$$e'_O = \text{CDs}(e_O + w_O, Enc(x_{1:T})),$$
$$p(y_{1:O}) = \text{Softmax}(Linear(e'_O)).$$

잠재 변수 z와 스타일 토큰 c는 코드 임베딩의 시퀀스 시작 부분에 추가된다. 이후 어텐션 블록은 z와 c에서 집계된 정보를 임베딩의 모든 프레임에 전달한다. CD의 나머지 부분은 타겟 코드를 재구성한다.

2.3 목적 함수

Style-VT의 주 목표는 음표 시퀀스 y의 주변 확률 분포를 근사하는 것이며, 이를 위해 Evidence Lower Bound (ELBO)를 최대화한다. Style-VT를 학습하기 위한 목적 함수는 다음과 같이 정의된다.

$$\mathcal{L}_{\text{Style-VT}} = \text{RE} + \lambda_S \text{SE} + \lambda_K \text{KLD}.$$

여기서 RE는 모델 출력과 목표 값 간의 차이를 측정하는 재구성 오류(Reconstruction Error)이며, 이는 다음과 같다.

$$\text{RE} = E_{q(z|x,y,c)}[-\log p_\theta(y \mid x, z, c)],$$

코드 확률 $p_\theta(y)$와 사후 분포 $q_\phi(z)$는 멜로디 입력 x와 스타일 토큰 c에 의해 조건화되며, 사전 분포 $p_\theta(z)$는 조건부 VAE 프레임워크에 따라 정규 분포를 따른다. SE는 대리 코드가 재구성 과정에 미치는 영향을 반영한 대체 재구성 오류(Substitution Reconstruction Error)로, 다음과 같다.

$$\text{SE} = -\sum_{i=1}^{N} \sum_{k=1}^{C} v_{yk} \log(p_{ik}),$$

여기서 N은 코드 시퀀스의 길이, C는 코드 집합을 나타낸다. v_y는 y 즉 목적 코드에 대한 대리 코드인지의 여부를 나타내는 크기 $|C|$의 벡터이다. 만약 $C_k \in \mathcal{S}(C_y)$이면, $v_{yk} = 1$이고, 그렇지 않으면 $v_{yk} = 0$이다. 벡터 p_i는 각 요소의 합이 1인 확률 벡터로, i번째 코드 시퀀스에 대해 모델이 예측한 확률을 나타낸다. KLD는 잠재 변수의 분포를 정규화 하는 쿨백-라이블러 발산으로, 다음과 같이 정의된다.

$$\text{KLD} = \text{KL}\left(q_\phi(z|x,y,c) \,\|\, p(z)\right).$$

Metric / Model	Harmonic Similarity			Diversity and Coherence					
	TPSD↓	DICD↓	LD↓	CHE↑	CC↑	CTR↑	PCS↑	CTD↓	MTD↓
GT	-	-	-	1.629	6.688	1.319	0.385	0.623	1.403
VTHarm	2.927	145.567	0.913	**1.9535**	8.0089	1.2120	0.3652	0.6853	1.4308
Style-VT (λ_s = 0)	2.905	136.156	0.863	1.9146	7.7889	1.2123	**0.3747**	**0.6559**	**1.4198**
Style-VT (λ_s = 0.1)	**2.833**	**131.133**	**0.847**	1.9529	**8.0533**	**1.2619**	0.3607	0.6961	1.4411

표 1. 화성 유사도, 코드의 다양성과 일관성 비교

3. 실 험

3.1 데이터 셋

실험을 위해 멜로디에 대한 코드 라벨이 음악의 장르별로 구성되어 있는 데이터셋을 구축하였으며, 음악의 장르는 재즈, 록, 기타의 세 가지 범주로 구성하였다. 데이터는 Chord Melody Dataset(CMD)의 데이터에 장르 레이블을 추가하는 방식으로 수집하였고, CMD 데이터셋의 장르 간 데이터 불균형을 해결하기 위해 MuseScore 악보 사이트로부터 재즈와 록 장르의 음악을 추가적으로 수집하였다. 모든 악보는 장조이며, 각 음악은 12 개의 조성으로 조옮김 되어있다. 데이터는 MusicXML 파일 형식이다. 사용되는 코드의 범위는 3 화음과 7 화음에서의 장화음, 단화음, 감화음이며, 12 개의 음계를 기반으로 총 72 개의 종류로 구성된다. 각 곡은 8 마디 단위로 훈련에 사용되고, 이전 훈련 단위와는 2 마디씩 중첩되어 나뉜다. 데이터셋은 train, validation, test 각각 8:1:1 비율로 나누었다. 데이터셋은 126 곡의 재즈 곡(91,845 test unit), 74 곡의 록 곡(80,369 test unit), 기타 장르 179 곡(109,271 test unit)으로 구성된다.

3.2 평가 지표 & 결과 분석

모델 평가를 위해 '화성 유사도'와 '코드의 다양성과 일관성' 두 가지의 지표를 사용하였다. '화성 유사도'는 생성된 코드와 인간이 작곡한 코드 사이의 화성 유사도로, [2]에서 제안된 세 가지 지표를 사용하여 측정하였다. Tonal Pitch Step Distance(TPSD), Interval Class Distance(DICD), Levenshtein Edit Distance(LD)의 세 가지 지표는 생성된 코드와 인간이 작곡한 코드 사이의 화성학적 유사도를 평가한다. '코드 다양성과 일관성'은 생성된 음악의 내부 구조를 평가하는 지표로, [3]에서 사용된 방법을 사용하였다. Chord Histogram Entropy(CHE)와 Chord Coverage(CC)는 코드의 다양성을 측정하며, Chord Tone to Non-chord Tone Ratio(CTR), Pitch Consonance Score(PCS), Melody-Chord Tonal Distance (MTD), 그리고 Chord Tonal Distance(CTD)는 멜로디와 코드, 코드와 코드 간의 일관성을 평가한다.

멜로디 기반 코드 생성 분야에서 가장 좋은 성능을 보이는 모델 중 하나인 VTHarm[3]을 베이스 라인으로 설정하였고, 인간이 작곡한 원곡의 코드 진행을 Ground-truth 로 설정하였다. 또한 대리 코드 이론의 적용이 성능에 미치는 영향을 평가하기 위해 Style-VT 는 각각 $\lambda_s = 0$, $\lambda_s = 0.1$ 로 설정된 두 개의 모델에 대하여 실험하였다. 평가 결과는 표 1 로 나타내었다. Style-VT($\lambda_s = 0.1$)는 화성 유사도를 측정하는 TPSD, DICD, LD 의 세 가지 지표 모두에서 가장 높은 성능을 보인다. 대리 코드 이론을 적용하지 않는 Style-V($\lambda_s = 0$) 또한 VTHarm 에 비해 우수한 성능을 보여준다. 이는 Style-VT 가 생성한 코드 진행이 인간이 작곡한 코드와 가장 유사하다는 것을 의미하며, 코드 생성에 있어 장르가 유의미한 정보를 가지고 있음을 반증한다. 코드 다양성 지표 CC 와 CHE 에서 Style-VT($\lambda_s = 0.1$)와 VTHarm 모델은 Style-VT($\lambda_s = 0$)모델에 비해 높은 값을 보인다. Style-VT($\lambda_s = 0$)은 장르 임베딩으로 코드에 대한 직접적인 정보를 받고, Style-VT($\lambda_s = 0.1$)에서는 대리 코드 이론의 적용으로 더 많은 선택지를 가지기 때문에 나타나는 결과로 해석된다. Style-VT($\lambda_s = 0$)은 PCS, CTD, MTD 에서 가장 높은 성능을 보이는데, 이는 장르의 임베딩이 멜로디와 코드 사이의 일관성 향상에 기여한다는 점을 보여준다.

4. 결 론

본 연구에서는 Transformer 와 VAE 아키텍처를 통합한 스타일 기반 코드 생성 모델인 Style-VT 를 제안하였다. 스타일 임베딩과 대리 코드 이론을 도입함으로써, Style-VT 는 장르별 코드 진행을 효과적으로 포착하였다. 이 연구는 스타일 기반 음악 생성에서의 향후 연구 방향을 제시하며, 다른 음악적 요소 및 여러 장르로의 확장을 가능하게 할 것이다.

참 고 문 헌

[1] I. Simon et al, "Mysong: automatic accompaniment generation for vocal melodies", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, p. 725-734, 2008.

[2] Y. C. Yeh et al, "Automatic melody harmonization with triad chords: A comparative study," Journal of New Mu sic Research, vol. 50, no. 1, pp. 37–51, 2021

[3] S. Rhyu, H. Choi, S. Kim, and K Lee, "Translating melody to chord: Structured and flexible harmonization of melody with transformer," IEEE Access, vol. 10, pp. 28261–28273, 2022.

[4] S. Wu et al "Melodyt5: Aunifiedscore-to-score transformer for symbolic music processing," arXiv:2407.02277, 2024.

[5] C.P´ erez-Sancho, D. Rizo, and J. M. I˜ nesta, "Genre classification using chords and stochastic language models," Connection Science, vol. 21, pp. 145–159, 2009.

[6] Y. Shih et al, "Theme transformer: Symbolic music generation with theme-conditioned transformer," IEEE Transactions on Multimedia, vol. 25, pp.3495–3508, 2022.

[7] A. Latham, Oxford Companion to Music, Oxford University Press, 1938.