JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 22, No. 6, pp. 907-913, Jun. 2021

Check for updates

딥러닝 기반 자동작곡에서 구성을 갖춘 곡 생성방법

정 석 환[1] · 정 성 훈[2*]
[1]한성대학교 지식서비스&컨설팅대학원 미래융합컨설팅학과 석사과정
[2*]한성대학교 기계전자공학부 교수

Generating Songs with Structure in Automatic Composition Based on Deep Learning

Suk-Hwan Chung[1] · Sung-Hoon Jung[2*]
[1]Master, Graduate School of Knowledge Service Consulting, Hansung University, Seoul 02876, Korea
[2*]Professor, School of Mechanical and Electronics Engineering, Hansung University, Seoul 02876, Korea

[요 약]

딥러닝 기반 자동작곡 방법이 많이 연구되고 있으나 곡의 구성을 갖춘 곡을 생성하는 방법은 거의 없는 실정이다. 이에 본 논문에서는 딥러닝 기반 자동작곡에서 곡 구성 정보를 함께 학습하여 구성을 갖춘 곡을 생성하는 방법을 제안한다. 곡의 진행에 따라서 동적으로 변하는 곡과 곡의 구성에 따라서 정적으로 주어지는 구성 정보를 함께 학습하기 위하여 동적인 정보와 정적인 정보를 함께 학습하는 딥러닝 모델을 사용하여 곡을 학습하였다. 제안한 방법을 이용하여 곡을 학습한 경우 곡 생성 시 곡의 구성 정보를 이용하여 곡을 생성할 수 있어서 보다 자연스러운 곡을 만들 수 있다. 생성한 곡을 평가하기 위하여 자연어처리에서 사용되는 METEOR과 BLEU를 사용하여 작곡가가 작곡한 곡과 얼마나 유사한지를 평가하는 방법으로 평가하였다. 평가 결과 곡의 구성을 적절히 주었을 때, 보다 더 작곡가가 작곡한 곡과 유사함을 볼 수 있었다.

[Abstract]

Although many deep learning based automatic music composition systems have been studied, it is hard to find a song generation system with the structure of songs. This paper addresses a method for generating songs with the structure of songs in deep learning-based automatic music generation by learning the structure together. We use a deep learning model that learns dynamic information which changes with the progress of the song and static information which is given according to the structure of the song together to learn the structure of the song. When a song is learned using the proposed method, it is possible to create a more natural song by using the composition information of the song when creating the song. We use METEOR and BLEU, which are used in natural language processing, to evaluate how similar the generated song is to a composer's. The results showed that they were more similar to the song created by a composer when a song was generated with the proper structure of songs.

색인어 : 자동작곡, 동적 데이터와 정적 데이터 결합, 곡 구성, METEOR, BLEU

Key word : Automatic composition, Combining dynamic data and static data, Song structure, METEOR, BLEU

http://dx.doi.org/10.9728/dcs.2021.22.6.907

Received 11 May 2021; **Revised** 23 June 2021
Accepted 23 June 2021

***Corresponding Author; Sung-Hoon Jung**

Tel: +82-2-760-4344
E-mail: shjung@hansung.ac.kr

Ⅰ. 서 론

최근 딥러닝 기법이 다양한 응용분야에 성공적으로 응용되고 있으며 음악 분야에서도 딥러닝으로 곡을 자동으로 생성하는 연구가 활발하게 진행되고 있다[1]-[3]. 딥러닝 자동작곡을 위하여 주로 사용되는 모델은 순환신경망[1]과 적대적 생성망[2]이 있다. 이러한 연구들로 인해 점점 더 자동작곡된 곡의 완성도 또한 높아지고 있다. 그러나 딥러닝 모델에 의하여 자동작곡된 곡들의 경우 멜로디의 느낌이 곡의 전체적인 구성과 어울리지 않게 생성되는 경우가 많다. 즉, 곡이 시작되는 부분에서 곡의 후렴부 느낌이 나거나 곡의 마무리 부분에서 곡의 도입부 느낌이 있거나 심지어 중간에 곡이 중단되었다가 다시 시작하는 경우도 있다. 이는 작곡을 위해서 딥러닝 모델을 학습시킬 때 단순히 곡의 멜로디만으로 학습하기 때문에 전반적인 곡 구성을 갖추지 못하기 때문이다.

이러한 문제를 해결하고자 본 논문에서는 곡을 학습하고 생성할 때 멜로디뿐만 아니라 곡의 구성도 함께 학습하여 주어진 곡의 구성에 따라서 생성할 수 있는 방법을 제안한다. 곡의 구성은 음악 장르에 따라서 다양하게 주어질 수 있는데, 예를 들면 가요의 경우 일반적으로 intro–verse–chorus–brigde–verse-chorus-outro 형태의 구성을 가진다. 그런데 곡의 구성은 몇 마디 동안 일정한 값을 갖기 때문에 해당 구간에서는 정적인 데이터가 된다. 그러므로 곡의 구성을 같이 학습하려면 동적인 멜로디와 정적인 구성을 같이 학습하는 방법을 사용해야 한다. 이런 다른 범주에 속하는 데이터를 결합하여 학습할 수 있는 방법에 대한 기존 연구가 있다[4]-[11]. M. Rahman, 등. (2020)은 시간이 지남에 따라 값이 변하는 동적인 시계열 데이터와 고정된 값을 가지는 정적 데이터를 동시에 같이 학습하는 방법으로 직접 입력 모델과 간접 입력 모델을 제안하였다[4]. 본 논문에서는 두 방식의 모델을 도입하여 동적인 곡과 정적인 구성 정보를 학습하여 결과를 비교하였다.

실험은 곡 구성이 비교적 단순하며 명확한 동요를 사용하여 수행하였으며 생성한 곡의 평가는 어떤 방식이 보다 더 작곡가가 작곡한 곡과 유사한 곡을 만드는지를 평가 기준으로 하여 평가하였다. 곡의 유사도를 비교하기 위한 방법으로는 자연어처리에서 번역이나 이미지 캡셔닝 등에서 성능평가로 자주 사용되는 METEOR (Metric for Evaluation of Translation with Explicit ORdering)과 BLEU (BiLingual Evaluation Understudy)로 평가하였다. 또한 평가의 공정성을 위하여 성능평가는 학습에 사용하지 않은 데이터인 테스트데이터를 이용하여 수행하였다. 곡 구성 정보가 실제로 곡 생성에 도움이 되는지를 확인하기 위하여 곡 생성 시에 다양한 방식의 구성 정보를 입력하여 곡을 생성한 결과를 비교하였다. 실험 결과 곡 구성 정보가 적절히 주어졌을 때 보다 더 작곡가가 작곡한 곡과 유사한 곡이 생성됨을 확인하였다. 그러나 동요의 경우 전체적으로 16마디 내외 수준으로 곡이 짧고 멜로디도 구성에 따라서 확실한 차이가 나지 않아서 성능 차이는 그렇게 크지 않았다. 향후 곡 정보가 더 복잡한 피아노 곡이나 K-POP 등에 적용해서 결과를 살펴볼 필요가 있다.

본 논문의 구성은 다음과 같다. 2절에서는 정적 데이터와 동적 데이터를 함께 학습시키는 방법과 생성된 곡 평가를 위한 METEOR과 BLEU에 관해서 설명한다. 3절에서는 본 논문에서 제안하는 멜로디와 곡 구성 데이터를 함께 학습시키는 구성을 갖춘 자동작곡 시스템에 관해서 설명한다. 4절에서는 제안한 방법으로 작곡한 결과를 평가하여 기술한다. 5절에서 결론으로 끝을 맺는다.

Ⅱ. 관련 연구

2-1 동적 정적 데이터 동시 학습 신경망

인공신경망을 응용하는 경우 동적인 시계열 데이터와 일정 시간 동안 특정한 값을 갖는 정적인 데이터를 같이 학습해야 하는 경우가 발생한다 [4-11]. 예를 들어 지역별 날씨 변화를 예측하는 인공신경망의 경우 동적 시계열 날씨 데이터와 함께 정적인 지역 정보를 같이 학습해야 한다. 자동작곡에서도 멜로디와 함께 곡의 구성을 학습하기 위해서는 이러한 방법을 사용해야 한다. 두 종류의 데이터를 같이 학습하기 위한 방법으로 두 가지 방식이 개발되었다 [4-11].

첫 번째는 동적 데이터를 순환신경망에 공급한 다음 정적 데이터와 연결하는 간접 입력(Indirect input) 방식이다 [4]-[8], [10]. 두 번째 접근방식은 정적 데이터와 동적 데이터를 순환신경망에 함께 입력하는 직접 입력(Direct input) 방식이다 [4], [5],[7]-[9],[11]. 그림 1은 간접 입력 방법과 직접 입력 방법의 구조를 보여준다. x_i는 정적 데이터 입력, x_t는 t단계에서의 동적 데이터 입력, y_{t+1}은 $t+1$ 단계에서의 예측을 나타낸다.

간접 입력 방식과 비교할 때 직접 입력 방식은 순환신경망 입력에 정적 데이터가 포함되기 때문에 정적 데이터의 비 시간적 정보로 인해 시간 특징 정보가 오염되는 단점이 있다[11].

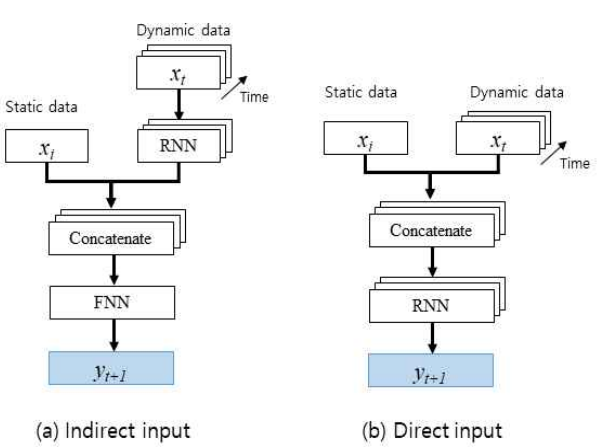

그림 1. 정적 데이터와 동적 데이터를 결합하는 두 가지 방법
Fig. 1. Two methods of integrating static and dynamic data

2-2 정량적 평가 방법

1) BLEU

BLEU는 기계가 번역한 내용의 품질을 평가하는 대표적인 알고리즘이다[12]. 기계가 번역한 문장을 인간이 번역한 문장들과 비교하여 번역의 품질을 평가한다. 0과 1 사이의 값을 가지며 1에 가까울수록 인간이 번역한 문장들과 더 유사하다는 것을 의미한다. BLEU는 n-gram을 기반으로 측정하며 언어에 구애받지 않고, 계산 속도가 빠르다는 장점이 있다.

기계가 번역한 문장을 Candidate이라고 하고 인간이 번역한 문장을 Reference라고 하자. BLEU 유니그램 정확도는 Candidate의 단어 중에 Reference에 등장한 단어의 개수를 세어 Candidate의 총 단어 수로 나눈 값이다. 이때 Candidate에 같은 단어가 반복적으로 나오는 경우가 있을 수 있으므로 Candidate를 구성하고 있는 단어들과 Reference를 구성하고 있는 단어들이 겹치는 수의 최댓값과 Candidate의 단어의 수 중 작은 수를 총 단어 수로 나누어 보정해 준다. 그러나 유니그램 정확도는 단어의 순서를 고려하지 않는다는 단점이 있다. 이를 해결하기 위해 유니그램에서 n-gram으로 확장하여 정확도를 계산한다. 보통 4-gram을 사용한다. 또한 Candidate의 길이가 짧은 경우에는 브레버티 패널티(Brevity Penalty)를 곱하여 정확도를 보정해 준다. 식 (1)은 브레버티 패널티를 나타낸다. 여기서 c는 Candidate의 길이, r은 Candidate와 가장 길이 차이가 작은 Reference의 길이를 말한다.

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp^{(1-\frac{c}{r})} & \text{if } c \le r \end{cases} \quad (1)$$

브레버티 패널티를 적용한 최종 BLEU는 식 (2)와 같다. p_n은 각 gram의 보정된 정확도를 나타낸다. N은 n의 최대 숫자이다. 보통 4의 값을 가진다. w_n은 각 gram의 보정된 정확도의 가중치를 말한다. 예를 들어 N이 4인 경우 p_1, p_2, p_3, p_4 에 대해서 동일한 가중치를 사용하면 0.25를 적용할 수 있다.

$$BLEU = BP \times \exp(\sum_{n=1}^{N} w_n \log p_n) \quad (2)$$

2) METEOR

METEOR은 BLEU와 마찬가지로 문장을 기본 단위로 하는 기계가 번역한 내용을 평가하기 위한 알고리즘이다[13].

기계가 번역한 문장을 Hypothesis라고 하고 인간이 번역한 문장을 Reference라고 하자. METEOR은 먼저 Hypothesis와 Reference 사이에 서로 일치하는 유니그램끼리 연결한 정렬을 만든다. 이때 Hypothesis의 유니그램은 Reference에 최대 1개의 유니그램과 연결될 수 있다. 이렇게 생성된 연결 정렬 중에 연결의 교차가 가장 적은 정렬이 선택된다. 선택된 정렬의 유니그램 정확도는 Reference에서 발견되는 Hypothesis의 단어 수를 Hypothesis의 단어 수로 나누어 구한다. 유니그램 재현율은 Reference에서 발견되는 Hypothesis의 단어 수를 Reference의 단어 수로 나누어 구한다. 이렇게 구한 정확도와 재현율을 조화평균을 사용하여 결합한다. 이때 재현율은 정확도보다 9배의 가중치를 준다.

유니그램을 n-gram으로 확장해보자. 일치하는 n-gram을 Reference와 Hypothesis 정렬에 사용하는 경우 패널티를 계산해 준다. 이때 Reference에서 Hypothesis와 일치하는 연속된 유니그램 세트, 즉 연결된 단어 묶음을 청크라 한다. Reference와 Hypothesis 사이에 청크 수가 더 적을수록, 즉 단어의 긴 연결이 많을수록 패널티는 낮아진다. 패널티 p는 식(3)과 같다. 여기서 c는 청크 수이고 u_m은 정렬된 유니그램의 수이다.

$$p = 0.5(\frac{c}{u_m})^3 \quad (3)$$

패널티를 적용한 최종 METEOR을 식 (4)에 나타내었다. 여기서 F_{mean}은 앞에서 구한 조화평균을 말한다.

$$M = F_{mean}(1-p) \quad (4)$$

Ⅲ. 연구 방법

3-1 데이터 전처리

곡을 인공신경망으로 처리하려면 컴퓨터가 이해할 수 있는 형태로 변경해 주어야 한다. 본 논문에서는 미디 파일의 곡 정보를 부호를 사용하여 텍스트로 변환하는 방식을 사용하였다.

1) 미디데이터

본 논문에서는 music21 파이썬 라이브러리를 사용하여 미디 파일 내의 음표, 쉼표, 음표 또는 쉼표의 길이, 박자 등의 정보를 추출하여 텍스트로 변환하는 방식을 사용하였다. 이 중 음표의 높낮이 정보는 music21에서 정의한 숫자 표기 방식을 따랐다. music21 라이브러리의 pitch.Pitch.midi 객체를 사용하면 음의 높낮이를 고정된 숫자로 나타낼 수 있다. 예를 들어 4옥타브 도(가온 다)는 숫자 60, 5옥타브 미는 숫자 76으로 표시된다. 쉼표는 rest를 의미하는 'r'로 나타내었고 높낮이를 가지고 있지 않으므로 길이만 표시 하였다. 박자표는 3/4, 4/4와 같이 나타내었다. 음표 또는 쉼표의 길이는 1.0과 같이 숫자로 나타내었다. 음표 또는 쉼표와 그 길이 사이는 '_'로 구분하였다. 음표 또는 쉼표의 길이와 다음 음표 또는 쉼표는 ','로 구분하였다. 마디와 마디 사이는 공백 문자를 사용하여 구분하였다. 마지막으로 박자표와 음표 또는 쉼표는 '|'로 구분하였다. 더불어 노래마다 다른 조성을 가지고 있으므로 조성에 따른 차이를 제거하기 위해서 모든 곡을 장조의 경우에는 다장조로, 단조의 경우에는 가단조로 조 옮김을 한 후 변환하였다.

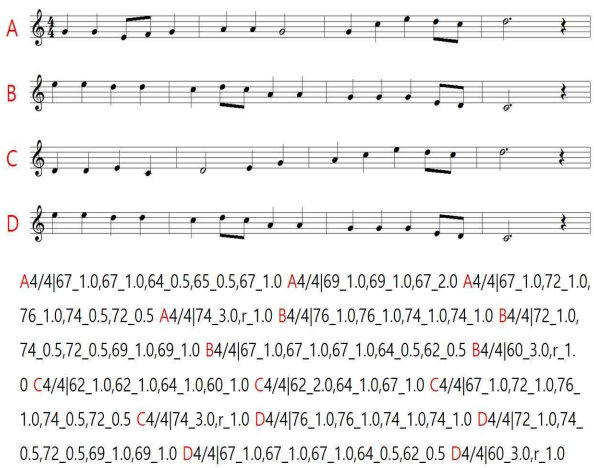

그림 2. 악보를 텍스트로 변환한 예
Fig. 2. An example of converting a music score to text

2) 곡 구성 정보

2-1절에서 언급한 바와 같이 음악은 멜로디 정보인 동적 데이터와 곡 구성 정보인 정적 데이터로 나눌 수 있다. 곡 구성 정보는 미디 파일에 포함되어 있지 않으므로 추가로 작업을 해주어야 한다. 본 논문에서는 곡 구성 정보를 넣어주기 위해서 비교적 쉬운 '기승전결'의 구조를 가지는 동요 300곡을 사용하였다. 300곡을 수작업으로 '기승전결'로 구분하여 각 파트를 A, B, C, D로 표시하였다. 일부 '기결(AD)' 또는 '기승결(ABD)'의 구조를 가지는 곡들도 포함되어 있다. 그림 2는 '고향의 봄'을 기승전결(A, B, C, D) 구조로 나누어 곡 구성 정보를 추가하여 미디 파일을 텍스트로 변환한 예를 보여준다.

3-2 자동작곡 시스템

이번 절에서는 인공신경망을 사용하여 작곡하는 시스템으로 3가지 모델을 제안한다.

1) 멜로디 학습 모델

첫 번째 모델은 자연어처리에서 자주 사용되는 순환신경망을 사용한 워드 임베딩 모델이다. 악보 데이터는 3-1절의 1)에서 설명한 데이터 전처리 방식을 사용하여 텍스트 형태로 변환된다. 변환된 텍스트 데이터는 마디 단위로 나뉘어 모델에 입력으로 들어간다. 입력된 마디는 마디 임베딩 층을 거쳐 벡터화된다. 순환신경망으로는 LSTM층 2개를 사용하였다. 출력단에서는 소프트맥스층을 거쳐 최종적으로 모델이 예측한 값이 생성된다. 입력한 정답과 비교 하여 손실을 계산하는 손실 함수로는 교차 엔트로피 오차 함수를 사용하였다. 그림 3 (a)는 마디 임베딩을 사용한 멜로디 학습 모델의 전체 구조를 보여준다.

2) 곡 구성 정보 직접 입력 모델

두 번째 모델은 멜로디 정보에 곡 생성을 위한 보조 데이터인 곡 구성 정보를 결합하여 순환신경망에 대한 단일 입력으로 넣어주는 방법이다. 악보 데이터는 3-1절의 2)에서 설명한 데이터 전처리 방식을 사용하여 텍스트 형태로 변환된다. 변환된 텍스트 데이터로부터 파트 정보와 멜로디 정보를 분리하여 결합층에 입력으로 넣어준다. 결합 층을 통과한 데이터는 마디 임베딩 층을 거쳐 벡터화된다. 순환신경망과 출력층은 멜로디 학습 모델과 동일하게 LSTM 층 2개와 소프트맥스층을 사용하였다. 손실 함수도 같은 교차 엔트로피 오차 함수를 사용하였다.

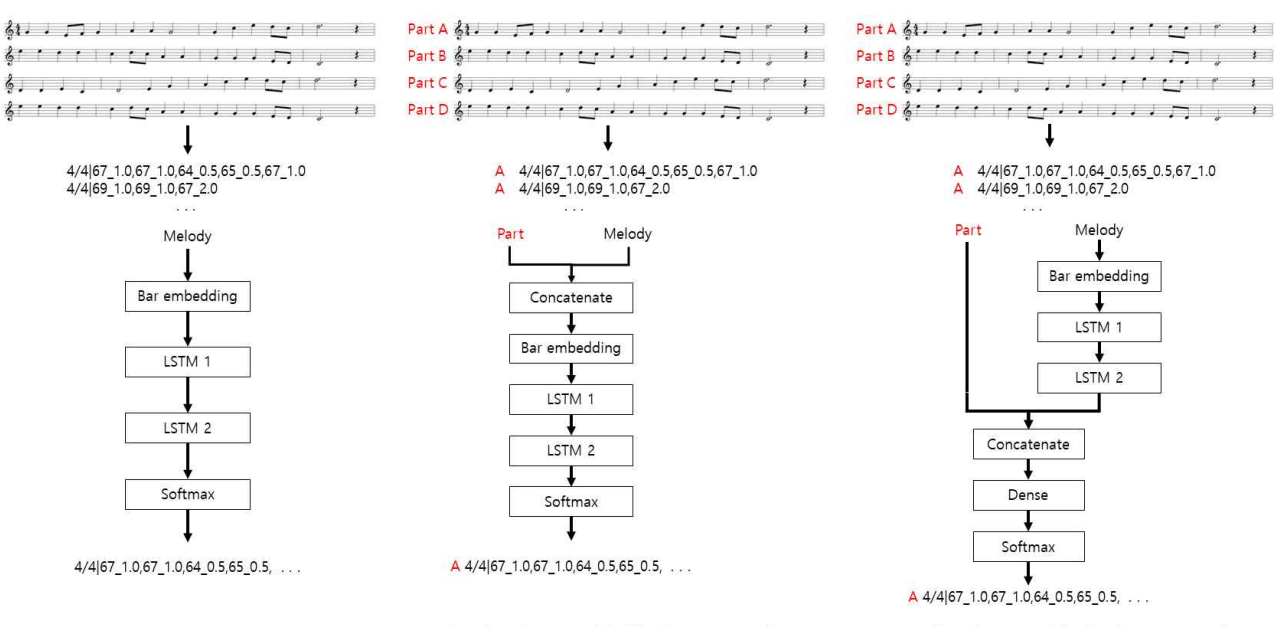

그림 3. 인공신경망을 사용한 자동작곡 모델
Fig. 3. Automatic song composition models

그림 3 (b)는 곡 구성 정보 직접 입력 모델의 전체 구조를 보여준다.

3) 곡 구성 정보 간접 입력 모델

마지막 모델은 곡 구성 정보를 별도로 사용하는 방식이다. 곡 구성 정보 직접 입력 모델과 동일하게 전처리 된 텍스트 데이터에서 멜로디 정보는 마디 임베딩 층을 거쳐 벡터화되고 2중 LSTM 층에 전달된다. LSTM 층을 통과한 데이터는 텍스트 데이터에서 분리된 파트 정보와 결합 된 후 피드포워드 신경망(FNN)을 거쳐 출력된다. 출력층과 손실 함수는 이전 모델들과 같게 소프트맥스층과 교차 엔트로피 오차 함수를 사용하였다. 그림 3 (c)는 곡 구성 정보 간접 입력 모델의 전체 구조를 보여준다.

Ⅳ. 실험결과

3절에서 제안한 각 모델을 사용하여 곡을 생성하고 분석하였다. 곡 구성 정보를 함께 학습한 모델에서는 곡을 생성할 때도 A-B-C-D의 곡 구성 정보를 함께 입력으로 넣어주어 곡을 생성하도록 하였다. 이렇게 생성된 곡들을 멜로디만 학습하여 생성된 곡들과 비교하였다. 일반적으로 동요는 16마디로 되어 있는 경우가 많으므로 모든 곡은 16마디로 생성하였다. 또한 학습한 적이 없는 D-C-B-A와 같은 순서의 곡 구성을 입력으로 주어 원하는 구성을 가지는 곡이 생성되는지를 실험하였다. 실험 데이터로는 동요 300곡이 사용되었다. 학습 훈련 데이터로 80%(240곡), 평가용 테스트데이터로 20%(60곡)를 사용하였다.

마지막으로 METEOR과 BLEU 점수를 사용하여 작곡가가 작곡한 곡과의 유사도를 모델별로 평가하여 보았다.

4-1 멜로디 학습 모델

구성 정보 없이 멜로디만 학습하는 멜로디 학습 모델에서는 학습 후 새로운 곡 생성 시 첫 마디만을 입력으로 주고 나머지 마디를 생성하도록 하였다. 이렇게 구성 정보 없이 생성한 곡에서 어느 정도 구성을 따르는지를 살펴보았다. 예상한 대로 구성 정보 없이 학습하고 생성하여 제대로 구성 정보를 따르지 않았다. 생성된 곡 중에 하나의 예를 들면, [AAB?C??DDDAAACC?]와 같이 생성되었다. 여기에서 ?는 해당 마디가 두 파트 이상에서 검색된 경우이다. 첫마디는 곡의 첫마디를 입력하여 A 파트이고 나머지는 거의 구성 정보와 상관없이 출력되는 것을 볼 수 있다. 이러한 결과는 본 논문의 제안 방법처럼 구성 정보를 같이 넣어서 학습하고 생성하는 것이 효과적임을 보여준다.

4-2 곡 구성 정보 입력 모델

곡 구성 정보 직접 입력 모델과 간접 입력 모델을 사용하여 멜로디와 곡 구성 정보를 순환신경망에 함께 입력하여 의도한 구성을 가진 곡이 생성되는지를 실험하였다. 동요에서 많이 사용되는 형식인 기승전결의 [AAAABBBBCCCCDDDD] 형식부터 일반적이지 않은 [DDDDCCCCBBBBAAAA] 형식 등 여러 형식을 멜로디와 결합하여(Concatenate) 순환신경망에 입력하여 곡을 생성하여 보았다. 표 1에 직접 입력 모델과 간접 입력 모델로 곡 생성 시 입력으로 사용된 곡 구성 정보와 이때 생성된 곡의 구성 정보의 예를 나타내었다. (<EOS>는 End of Song의 약자로 곡의 끝부분을 나타낸다. 자연어처리에서 End of Sentence와 같은 구분자 역할을 한다)

[AAAABBBBCCCCDDDD] 형식으로 생성한 곡은 입력해준 곡 구성 정보와 거의 동일한 구성을 하고 있음을 볼 수 있다. 이는 학습할 때와 동일한 구성 정보로서 학습이 잘 수행되었음을 나타낸다. 간접 입력 모델의 경우 입력한 구성 정보와 완전히 동일한 구성을 갖는 곡을 생성하는 데 반하여, 직접 입력 모델은 입력한 구성 정보와 약간 다른 구성을 보임을 알 수 있다. 이는 직접 입력 모델에서는 시간에 따라 변하는 멜로디와 같이 입력하여 데이터가 더 변형되기 쉬운 것 때문으로 판단된다.

그 이외에 학습할 때와 다른 구성 정보에서 어떻게 곡을 생성하는지 보기 위하여 여러 가지 다른 구성 정보를 넣어서 곡을 생성하여 보았다. 기승전결이 빠르게 반복되는 [AABBCCDDAABBCCDD] 형식으로 생성한 곡은 어느 정도 입력 구성을 따라가는 것을 볼 수 있다. 구성이 느리게 변하는 [AAAAAAAABBBBBBBB] 형식으로 생성한 곡은 뒷부분으로 갈수록 생성한 곡의 구성이 입력된 구성과 달라짐을 볼 수 있다. 이는 순환신경망의 특성상 시간이 흐를수록 입력된 정보가 약해지는 현상으로 설명된다. 구성이 변하지 않는 [AAAAAAAAAAAAAAAA] 형식으로 생성한 곡 역시 시간이 흐를수록 입력된 마디 정보가 약해지는 현상을 보인다.

마지막으로 일반적인지 않은 [DDDDCCCCBBBBAAAA] 형식으로 생성한 곡의 마디는 특별한 형식을 가지고 있지 않다고 볼 수 있다. 이는 단방향 순환신경망을 사용하였기 때문에 역방향 구성을 가지는 곡을 생성하지 못하고 있다. 이러한 결과들을 보았을 때 곡의 구성에 따라서 학습하고 생성할 때도 학습한 곡의 구성과 유사하게 구성을 넣어주면 대부분 곡의 구성을 맞추어 생성할 수 있음을 보여준다.

표 1. 곡 구성 정보 입력 모델로 생성한 곡의 구성 정보
Table 1. Structure of generated songs by direct and indirect input models

Input structure of song	Structure of generated songs by direct input model	Structure of generated songs by indirect input model
AAAABBBB CCCCDDDD	AAADBBBB CCCCDDDD	AAAABBBB CCCCDDDD
AABBCCDD AABBCCDD	AACACBBC CDDD<EOS>	AACAAABB <EOS>ACDDDBD
AAAAAAAA BBBBBBBB	AAAAAABB <EOS>DBBCCDB	AAAAA<EOS>BA ACCCDBDD
AAAAAAAA AAAAAAAA	AAAAABBB BBADCBD	AAABABBC CCCCDCC
DDDDCCCC BBBBAAAA	DCBBAAAA CBBBCCAC	DCBBABAD ADDDDCCD

4-3 곡 평가

자동작곡 모델의 품질을 평가하는 데 있어서 중요한 것 중 하나는, 생성된 음악이 궁극적으로 얼마나 사람들이 듣기에 좋은가 하는 것이다. 음악 평가는 주관적이고 사람에 따라 다르므로 정확한 평가가 어렵다는 문제가 있다. 이를 해결하기 위해 자연어처리 모델에서 사용되는 METEOR과 BLEU와 점수를 사용하여 인공신경망에 의해 생성된 곡이 작곡가가 작곡한 곡과 얼마나 유사한지를 평가하였다. METEOR과 BLEU 점수는 모두 값이 1에 가까울수록 성능이 우수함을 나타낸다. BLEU 점수 측정 시에 n-gram의 수는 4로 기본 가중치 (0.25, 0.25, 0.25, 0.25)를 사용하였다.

1) 모델별 평가

곡 학습과 생성에 사용된 세 가지 모델을 비교하기 위하여 모델별로 [AAAABBBBCCCCDDDD] 구성으로 100곡씩 생성하여 METEOR과 BLEU 점수를 측정하였다. 표 2는 각 학습 모델이 생성한 100곡에 대한 METEOR과 BLEU 점수의 평균을 나타낸다. 곡 구성 정보를 추가한 모델들이 멜로디만 학습한 모델에 비해 METEOR과 BLUE 모두 점수가 높게 나왔다. 이는 신경망 학습 시 곡 구성 정보를 함께 학습하여 생성된 곡이 작곡가가 작곡한 곡에 더 가깝다는 것을 보여준다.

직접 입력 모델과 간접 입력 모델을 비교하였을 때는 간접 입력 모델이 METEOR의 경우 약 0.022(6.1%), BLEU의 경우 약 0.019(4.2%) 정도 높은 점수가 나왔다. 2-1절에서 언급한 것 같이 직접 입력 방법은 순환신경망 초기 단계에서 정적 데이터를 함께 포함하게 되므로 순환신경망으로 들어가는 시계열 데이터를 오염시키는 단점이 있다. 이로 인해 간접 입력 모델이 직접 입력 모델보다는 작곡가가 작곡한 곡에 조금 더 가깝다는 것을 알 수 있다.

2) 곡 구성 방식별 평가

표 3은 직접 입력 모델과 간접 입력 모델로 평가한 여러 구성 형태에 따른 METEOR 과 BLEU 점수를 보여준다. 두 모델 모두 원곡과 같거나 비슷한 형태를 가진 구성으로 곡을 생성하였을 때 METEOR과 BLEU 모두 점수가 높게 나오는 것을 볼 수 있다. 예를 들어 표 3의 간접 입력 모델의 METEOR 점수를 보면 [AAAABBBBCCCCDDDD], [AAAAAAAABBBBBBBB], [AABBCCDDAABBCCDD], [AAAAAAAAAAAAAAAA] 순으로 점수가 높다. 이는 원곡들의 구성인 [AAAABBBBCCCCDDDD]와 비슷한 형식을 가진 곡들이 높은 점수를 받은 것이다.

BLEU 평가의 경우 점수들이 좁은 구간에 몰려있다. 예를 들어 표 3의 간접 입력 모델의 BLEU의 가장 높은 점수와 낮은 점수의 차이는 0.011 (0.477-0.466 = 0.011) 밖에 나지 않는다. 이에 비하여 표 3의 간접 입력 모델의 METEOR의 가장 높은 점수와 낮은 점수의 차이는 0.082 (0.388-0.306 = 0.082)이다. 이는 METEOR이 BLEU와는 다르게 곡을 평가할 때 생성된 마디들의 존재 여부뿐만 아니라 순서까지도 고려하기 때문이다.

표 2. 각 모델로 생성한 100곡의 METEOR과 BLUE 평균 점수
Table 2. Mean of METEOR and BLEU scores for 100 songs generated by each model

	Melody only	Direct input model	Indirect input model
METEOR	0.257	0.365	0.388
BLEU	0.454	0.458	0.477

표 3. 곡 구성에 따른 METEOR과 BLEU 점수
Table 3. METEOR and BLEU scores for the input structure of songs

Structure of Song (label)	METEOR (direct input)	BLEU (direct input)	METEOR (indirect input)	BLEU (indirect input)
AAAAAAAA AAAAAAAA	0.346	0.467	0.366	0.467
BBBBBBBB BBBBBBBB	0.342	0.467	0.323	0.466
CCCCCCCC CCCCCCCC	0.317	0.462	0.306	0.469
DDDDDDDD DDDDDDDD	0.315	0.462	0.313	0.468
AAAAAAAA BBBBBBBB	0.357	0.472	0.372	0.471
CCCCCCCC DDDDDDDD	0.311	0.453	0.311	0.469
AAAABBBB CCCCDDDD	0.365	0.518	0.388	0.477
AABBCCDD AABBCCDD	0.348	0.487	0.368	0.473
DDDDCCCC BBBBAAAA	0.344	0.434	0.360	0.471

Ⅴ. 결 론

본 논문에서 우리는 딥러닝을 이용한 자동작곡에서 곡 생성 시 구성을 갖춘 곡을 생성하기 위하여 멜로디와 함께 곡의 구성을 학습하고 생성 시에 구성 정보를 이용하여 곡을 생성하는 방법을 제안하였다. 동적인 멜로디와 정적인 곡 구성을 함께 학습하기 위하여 정적 정보와 동적 정보를 함께 학습할 수 있는 두 가지 인공신경망 모델을 이용하여 실험하였다. 생성된 곡을 평가하기 위하여 METROR 과 BLEU를 사용하여 생성된 곡과 작곡가가 만든 곡의 유사도를 평가하였다. 실험 결과 곡의 구성을 이용하여 학습하고 생성한 것이 멜로디만 학습하여 생성한 것보다 작곡가가 작곡한 곡과 더 유사함을 볼 수 있었다. 또한 직접 입력 모델보다 간접 입력 모델이 더 작곡가가 작곡한 곡과 유사하였다. 특히, 학습 시에 넣어준 구성과 동일하게 구성을 입력하여 생성한 곡이 가장 작곡가가 작곡한 곡과 유사하였다. 이를 통해 딥러닝을 이용한 자동작곡에서 곡 구성 정보를 이용하여 학습하고 생성하는 것이 더 좋은 곡을 생성할 수 있음을 보았다. 결과적으로 본 논문에서 제안한 방법은 곡 구성을 갖춘 곡을 생성할 수 있으며 다양한 구성에 따라서 다양한 곡을 생성할 수 있는 장점이 있다. 향후 곡의 길이가 긴 가요나 팝송에 적용하여 결과를 확인하는 것이 필요하다.

감사의 글

본 연구는 한성대학교 교내학술연구비 지원과제입니다.

참고문헌

[1] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, “Music transcription modelling and composition using deep learning”, arXiv:1604.08723, 2016

[2] H. W. Dong, W. Y. Hsiao, L. C. Yang, and Y. H. Yang, “MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment”, arXiv:1709.06298, 2018

[3] K. Nikhil. “Bach2Bach: Generating Music Using A Deep Reinforcement Learning Approach”, arXiv:1812.01060, 2018

[4] M. H. Rahman, S. Yuan, C. Xie, and Z. Sha, “Predicting human design decisions with deep recurrent neural network combining static and dynamic data”, Design Science. 6. 10.1017/dsj, 2020

[5] J. T. Kristensen and P. Burelli, "Combining Sequential and Aggregated Data for Churn Prediction in Casual Freemium Games," 2019 IEEE Conference on Games (CoG), London, UK, pp. 1-8, 2019

[6] F. Zhu, X. Song, C. Zhong, S. Fang, R. Bouchard, V. N. Fontama, P. Singh, J. Gao and L. Deng, “Churn prediction using static and dynamic features”, US Patent App. 15/446,870, 2018

[7] A. Leontjeva and I. Kuzovkin, "Combining Static and Dynamic Features for Multivariate Sequence Classification," 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, pp. 21-30, 2016

[8] C. Lin, Y. Zhangy, J. Ivy, M. Capan, R. Arnold, J. Huddleston, M. Chi, “Early diagnosis and prediction of sepsis shock by combining static and dynamic information using convolutional-lstm”, 2018 IEEE International Conference on Healthcare Informatics (ICHI), 219-228 (IEEE), 2018

[9] C. Esteban, O. Staeck, Y. Yang, and V. Tresp, “Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks”, 2016 IEEE International Conference on Healthcare Informatics (ICHI), 93-101 (IEEE), 2016

[10] T. C. Hsu, S. T. Liou, Y. P. Wang, Y. S. Huang, et al., “Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction”, ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1572-1576 (IEEE), 2019

[11] T. Wang, F. Jin, Y. Hu, and Y. Cheng, “Early Predictions for Medical Crowdfunding: A Deep Learning Approach Using Diverse Inputs”, arXiv:1911.05702, 2019.

[12] S. Banerjee, and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”, In Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association of Computational Linguistics, Ann Arbor, Michigan, June 2005

[13] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: a method for automatic evaluation of machine translation”, ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318. CiteSeerX 10.1.1.19.9416, 2002

정석환(Suk-Hwan Chung)

2001년 : 중앙대학교 (공학사)
2021년 : 한성대학교 (컨설팅학석사)

2003년～2012년: Motorola Korea
2013년～현 재: NXP Semiconductors
※관심분야 : 인공지능, 보안 등

정성훈(Sung-Hoon Jung)

1988년 : 한양대학교 (공학사)
1991년 : KAIST (공학석사)
1995년 : KAIST (공학박사)

1996년～현 재: 한성대학교 기계전자공학부 교수
※관심분야 : 인공지능, 시스템생물학, 융합공학 등