KIPS Trans. Softw. and Data Eng.
Vol.10, No.11 pp.483~490
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2021.10.11.483

Creating Songs Using Note Embedding and Bar Embedding and Quantitatively Evaluating Methods

Young-Bae Lee† · Sung Hoon Jung††

ABSTRACT

In order to learn an existing song and create a new song using an artificial neural network, it is necessary to convert the song into numerical data that the neural network can recognize as a preprocessing process, and one-hot encoding has been used until now. In this paper, we proposed a note embedding method using notes as a basic unit and a bar embedding method that uses the bar as the basic unit, and compared the performance with the existing one-hot encoding. The performance comparison was conducted based on quantitative evaluation to determine which method produced a song more similar to the song composed by the composer, and quantitative evaluation methods used in the field of natural language processing were used as the evaluation method. As a result of the evaluation, the song created with bar embedding was the best, followed by note embedding. This is significant in that the note embedding and bar embedding proposed in this paper create a song that is more similar to the song composed by the composer than the existing one-hot encoding.

Keywords : Automatic Composition, One-Hot Encoding, Note Embedding, Bar Embedding, Quantitative Evaluation

음표 임베딩과 마디 임베딩을 이용한 곡의 생성 및 정량적 평가 방법

이 영 배† · 정 성 훈††

요 약

인공신경망을 이용해서 기존 곡을 학습시키고 새로운 곡을 생성하기 위해서는 전처리 과정으로 곡을 신경망이 인식할 수 있는 숫자로 변환해야 하며, 지금까지는 원-핫 인코딩이 사용되어 왔다. 본 논문에서는 음표 임베딩과 마디 임베딩을 제안하고 기존의 원-핫 인코딩과 성능을 비교하였다. 성능비교는 어떤 방식이 작곡가가 작곡한 곡과 유사한 곡을 생성하는지를 정량적 평가에 근거해서 수행하였으며, 평가방법으로는 자연어 처리 분야에서 사용되는 정량적 평가 방법들을 이용하였다. 평가결과 마디 임베딩으로 생성한 곡이 가장 좋았으며 그 다음으로 음표 임베딩이 좋았다. 이는 본 논문에서 제안한 음표 임베딩과 마디 임베딩이 원-핫 인코딩보다 작곡가가 작곡한 곡과 유사한 곡을 생성한 것으로서 의의가 있다.

키워드 : 자동 작곡, 원-핫 인코팅, 음표 임베딩, 마디 임베딩, 정량적 평가

1. 서 론

인공신경망을 사용하여 곡을 학습시키고 생성하기 위해서는 먼저 전처리 과정으로 음표와 쉼표, 박자와 같은 데이터를 인공 신경망이 인식할 수 있는 수치 데이터의 형태로 바꿔야 한다. 이러한 전처리 방식으로는 0과 1로 이루어진 희소 벡터로 변환시켜주는 원-핫 인코딩이 사용되어 왔다[1-4]. 그러나 원-핫 인코딩은 데이터를 희소 벡터의 형태로 변환시키기 때문에 데이터간의 관계나 의미를 전혀 반영하지 못하는 문제가 있다[5].

우리는 이러한 문제점을 완화하기 위하여 본 논문에서 곡의 음표와 마디 수준에서 임베딩을 하는 음표 임베딩과 마디 임베딩을 제안한다. 음표 임베딩이라 함은 곡을 구성하고 있는 음표와 쉼표를 기본 단위로 음의 길이와 높이, 그리고 쉼표의 길이를 밀집 벡터로 표현하는 것이다. 음표와 쉼표를 희소 벡터가 아니라 밀집 벡터로 표현함으로써 학습 데이터의 차원을 축소할 수 있고, 각 음표의 앞뒤 쓰임새에 따라서 벡터화하기 때문에 원-핫 인코딩과는 달리 임베딩 된 벡터에 맥락 정보를 포함할 수 있다. 이는 자연어 처리에서 단어를 원-핫 인코딩과 단어 임베딩으로 처리하는 경우의 차이라고 할 수 있다[5]. 마디 임베딩은 곡의 마디 수준에서 임베딩 하

※ 본 연구는 한성대학교 교내학술연구비 지원과제임.
† 비 회 원 : 한성대학교 지식서비스&컨설팅대학원 미래융합컨설팅학과 석사
†† 정 회 원 : 한성대학교 기계전자공학부 교수
Manuscript Received : September 30, 2021
Accepted : October 18, 2021
* Corresponding Author : Sung Hoon Jung(shjung@hansung.ac.kr)

는 것으로서 음표 임베딩을 자연어처리에서 단어 임베딩으로 본다면 마디 임베딩은 이러한 단어가 모인 문장을 단위로 임베딩하는 문장 임베딩으로 볼 수 있다. 즉 마디 임베딩의 경우 마디를 구성하고 있는 모든 음표와 쉼표의 종류와 길이에 따라서 하나의 마디를 하나의 벡터로 표현하는 것이다. 그러므로 음표에 비해 종류가 훨씬 더 다양하기 때문에 음표 임베딩보다 곡의 맥락을 충분히 임베딩 할 수 있을 것이다. 제안한 음표 임베딩과 마디 임베딩의 효과를 보여주기 위하여 기존의 원-핫 인코딩과 비교 실험을 하였다.

곡을 평가하는 방법에는 다수의 사람이 주관적으로 평가한 것을 통계 내어 평가하는 정성적 평가 방법이 있다. 그러나 음악이 가지는 창의성이나 아름다움, 예술적 가치를 평가할 수 있는 평가 지표를 개발하는 것 그리고 편향성을 제거하며 충분한 수의 자격 있는 피험자들을 모집하는 것은 매우 어렵고 비용도 많이 든다[6,7]. 결과적으로 정성적 평가는 사람의 주관성이 커서 객관적 정량적 평가에는 한계가 있으며, 곡에는 정답이 없기 때문에 곡 자체에 대한 객관적 정량적 평가에는 많은 어려움이 존재한다.

우리는 이러한 문제점을 해결하고자 작곡가가 작곡한 곡을 정답으로 보고 얼마나 작곡가가 작곡한 곡과 유사한 곡을 생성하는지를 판단하는 방법으로 각 전처리 방법을 평가하는 것으로 결정하였다. 그러나 자동작곡 분야의 연구에서 두 곡의 유사도를 객관적인 평가 지표에 근거해서 평가하는 방법이 개발된 것이 없었다. 그렇지만 자동작곡과 유사하게 시계열 데이터를 다루는 자연어처리 분야에서는 유사도를 측정하는 방법이 개발되어 있다[8-11]. 자연어 처리 분야에서 사용되는 정량적 평가 방법들은 번역 모델이나 텍스트 요약모델이 생성한 문장이 인간이 번역하거나 요약한 문장과 얼마나 유사한지를 평가한다[8-10]. 본 논문에서는 이렇게 자연어처리에서 개발된 유사도 평가 방법인 BLEU[8], ROUGE[9], METEOR[10]를 사용하여 평가하였다.

각 방법을 객관적 정량적으로 비교하기 위하여 임베딩 방법 외에 다른 조건들은 동일하게 실험을 수행하였다. 전체 데이터에서 학습용 데이터로 학습시키고 성능 비교는 나머지 평가용 데이터로 수행하였다. 마디 임베딩의 경우 마디 단위로 학습하고 생성된 곡이 마디 단위로 출력되어서 유리한 측면이 있지만 성능비교는 학습에 사용하지 않은 평가용 데이터를 이용하였기 때문에 공평성에 문제가 없다. 마디 임베딩의 경우는 자동작곡한 곡이 마디 수준의 벡터이기 때문에 작곡가가 작곡한 곡과의 유사도를 비교하기 위하여 마디벡터를 음표로 변환하여 평가하였다. 실험결과 동일한 조건에서는 마디 임베딩이 음표 임베딩보다 더 높은 평가 점수를 얻었으며, 음표 임베딩은 원-핫 인코딩보다 항상 더 높은 평가 점수를 얻었다. 이를 통해 마디 임베딩이나 음표 임베딩이 기존의 원-핫 인코딩보다 작곡가가 작곡한 곡과 더 유사한 곡을 생성한다는 것을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2절에서 기존의 정량적 평가 방법과 자연어 처리 분야에서 사용되는 정량적 평가 방법들에 대하여 소개한다. 3절에서는 전처리 과정과 정량적 평가에 사용할 평가 방법을 기술한다. 4절에서는 실험결과에 대하여 설명하며 5절의 결론으로 끝을 맺는다.

2. 관련 연구

2.1 기존의 정량적 평가 방법

학습에 사용된 작곡가가 작곡한 곡과 자동 작곡 모델에 의하여 생성된 곡의 유사도를 통계적 기법으로 비교하는 방법이 연구 되었다[6]. 이 방법에서는 먼저 피치 기반 특성들과 노트 기반 특성들을 설정하고 학습에 사용된 곡으로 구성된 데이터 세트와 자동 작곡 모델에 의해 생성된 곡으로 구성된 데이터 세트를 준비한다. 그 후에 두 데이터 세트로부터 특성들을 추출한 후, 커널 밀도 추정을 이용하여 각 특성의 확률 밀도 함수를 추정한다. 확률 밀도 함수 그래프 간에 서로 겹치는 정도를 의미하는 중복 영역과 확률분포의 차이를 의미하는 쿨백-라이블러 발산을 정량적 평가의 지표로 설정하여 평가한다. 확률 밀도 함수 그래프 간에 중복 영역이 크고 쿨백-라이블러 발산이 작을수록 모델에 의해 생성된 곡과 학습에 사용된 곡은 유사하다고 평가한다[6].

2.2 자연어 처리 분야에서 사용되는 정량적 평가 방법

1) BLEU

후보 문장과 참조 문장 사이의 n-그램을 비교하여 수정된 정밀도를 구하고, 지나치게 짧은 문장이 생성될 때 이를 개선하기 위해서 브레버티 페널티(brevity penalty, BP)를 적용하여 다음과 같이 BLEU-N 점수를 구한다[8,12].

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

여기서 w_n은 n-gram간의 가중치로서 일반적으로 동일하게 설정하고, p_n은 n-gram 정확도(precision)이다.

2) ROUGE

텍스트 요약 모델이 생성한 요약본을 사람이 미리 만들어 놓은 참조 요약본과 비교하여 정확도(precision)와 재현율(recall), 두 값의 F-measure를 구하여 해당 텍스트 요약 모델의 성능을 평가한다[4]. n-그램을 기반으로 하는 ROUGE-N과 LCS (longest common subsequence)를 기반으로 하는 ROUGE-L 등이 있다[9].

3) METEOR

METEOR 점수를 계산하기 위해서는 먼저 후보 문장의 각 유니 그램을 참조 문장의 유니 그램과 연결하여 후보 문장과 참조 문장 사이의 정렬을 만들고, 후보 문장과 참조 문장 사

이의 정렬을 기반으로 유니 그램 정밀도와 재현율, 두 값의 F-measure를 구하여 다음과 같이 METEOR 점수를 구한다[10].

$$\text{METEOR 점수} = \frac{10RP}{R+9P} \times (1 - \text{Penalty}) \quad (2)$$

여기서 P는 정확도(precision)이고, R은 재현율(recall)이며, Penalty는 후보 문장과 참조 문장 간의 유사도를 반영하는 개념으로서 두 문장이 유사할수록 Penalty는 작아지고 두 문장 간의 유사도가 낮아질수록 Penalty는 커진다.

4) CIDEr

이미지 캡셔닝(image captioning) 생성 모델에 대한 정량적인 평가를 위해서 제안된 것으로, 학습 데이터 세트를 구성하고 있는 이미지 주석에서 자주 등장하는 단어는 불용어(stopword)일 확률이 높기 때문에 입력 이미지에 대해서 이미지 캡셔닝 모델이 생성한 주석에 학습 데이터 세트에서 자주 등장하는 단어가 많이 포함될수록 TF-IDF(term frequency inverse document frequency)를 사용하여 모델의 성능을 낮게 평가한다[11].

3. 연구 방법

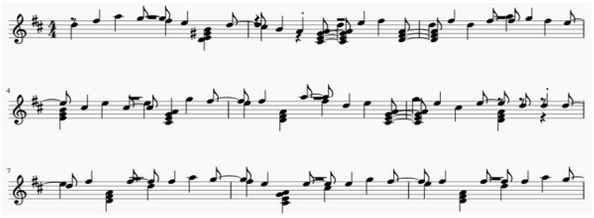
3.1 전처리 과정

인공신경망을 이용해서 음악을 학습시키고, 새로운 곡을 생성하기 위해서는 먼저 곡을 구성하고 있는 음악 데이터를 순환 신경망이 인식하고 처리할 수 있는 형태로 바꿔줘야 한다. 본 논문에서는 전처리 과정을 2단계로 진행하며, 첫 번째 단계는 미디 파일 형식의 곡 정보를 텍스트로 변환하며, 두 번째 단계에서는 변환된 텍스트를 수치 데이터로 변환한다.

1) 미디 파일을 텍스트 데이터로 변환

우리는 파이썬 라이브러리 중에서 music21을 이용하여 미디 파일 내의 다양한 음악 데이터를 다음과 같이 텍스트로 변환하였다.

a) 음의 높낮이는 music21 pitch 클래스의 midi라는 속성을 사용하여 music21에서 정의하고 있는 숫자를 부여한다. (예) 4옥타브 도(c4)는 60, 5옥타브 도(c5)는 72
b) 쉼표(rest)에 대해서는 자체로 정한 독립적인 기호 r을 부여한다.
c) 음표와 쉼표의 길이는 우리가 설정한 숫자를 부여한다. (예) 8분 음표는 0.5, 4분 음표는 1.0, 4분 쉼표는 1.0
d) 음의 높낮이, 음표 및 쉼표의 길이는 '_'로 이어서 표현한다. (예) 60_0.5는 C4 8분 음표, 72_1.0은 C5 4분 음표
e) 화음은 별표(*)를 이용하여 단음을 연결한다. (예) 60*72_0.5는 C4 C5 16분 음표 2개가 연결된 화음
f) 단음, 쉼표, 화음은 쉼표(,)를 이용하여 구분한다. (예) r_3.0,60*61_0.5,60_0.5



74_1.0 78_1.0 81_1.0 79_1.0 78_1.0 76_1.0 68*71*62*64_1.0 74_1.0 73_1.0 71_1.0 69_0.5 61*64*67_1.0 74_1.0
76_1.0 78_1.0 62*66*69_1.0 74_1.0 78_1.0 74_1.0 78_1.0 81_1.0 79_1.0 78_1.0 76_1.0 64*67*71_1.0 73_1.0
76_1.0 73_1.0 76_1.0 61*64*67*69_1.0 79_1.0 78_1.0 76_1.0 78_1.0 62*66*69_1.0 81_1.0 78_1.0 74_1.0 76_1.0
61*64*67*69_1.0 79_1.0 76_1.0 73_1.0 76_1.0 62*66*69_1.0 74_0.5 r_0.5 74_0.5 76_1.0 78_1.0 62*66*69_1.0
78_1.0 74_1.0 78_1.0 81_1.0 79_1.0 78_1.0 76_1.0 61*64*67*69_1.0 76_1.0 73_1.0 76_1.0 79_1.0 78_1.0 76_1.0

Fig. 1. An Example of Converting a Midi File to Text

g) 박자표 뒤에 '|'를 붙여서 음표 및 쉼표와 구분한다.
h) 각 마디들은 공백 문자 ' '로 구분한다. (예) 3/4|r_2.0,60_1.0 3/4|61_2.0,60_1.0

Fig. 1은 미디 파일을 텍스트로 변환한 예이다.

2) 텍스트 데이터를 수치 데이터로 변환

기존에는 원-핫 인코딩 방식을 사용하였으나 우리는 음표와 쉼표를 기본 단위로 하는 음표 임베딩, 그리고 악보의 마디를 기본 단위로 하는 마디 임베딩을 새롭게 구현하였다.

a) 원-핫 인코딩: 원-핫 인코딩[1-4]은 음표 및 쉼표와 같은 음악 데이터들에 대해서 고유한 인덱스를 부여하고 음악 데이터의 종류에 해당하는 숫자를 벡터의 크기로 설정한다. 표현하고 싶은 음악 데이터의 인덱스에는 1, 나머지 인덱스에는 0을 부여하는 방식이며, 이렇게 생성된 벡터를 원-핫 벡터라고 한다. 원-핫 인코딩 방식은 텍스트를 숫자로 변환시키지만 1과 0으로 채워진 희소 벡터를 생성하기 때문에, 생성된 벡터들 간에 덧셈, 뺄셈 및 코사인 유사도와 같은 어떤 연산도 불가능하다. 이로 인해서 원-핫 인코딩 방식은 음악 데이터들 간에 어떤 관계성도 부여할 수 없다.
b) 음표 임베딩: 자동작곡에서 음표 임베딩은 자연어처리에서 단어 임베딩과 그리고 마디 임베딩은 자연어처리에서 문장 임베딩과 유사하다고 볼 수 있다. 그러므로 음표 임베딩이나 마디 임베딩의 특징은 자연어처리에서 단어 임베딩의 특징과 유사하다. 단어 임베딩 방식은 원-핫 인코딩 방식과는 다르게 밀집 벡터를 생성하므로 표현하고자 하는 단어의 수보다 훨씬 적은 차원의 벡터로 학습에 사용될 모든 단어들을 표현할 수 있다. 뿐만 아니라 생성된 임베딩 벡터들 간에는 코사인 유사도와 같은 벡터간의 연산이 가능하다. 이로 인해서 원-핫 인코딩과는 다르게 각 단어에 관계성을 부여할 수 있으며, 이를 통해서 단어 사이의 연관성이나 문맥을 활용할 수 있게 되었다[13-15].
c) 마디 임베딩 : 우리는 다음과 같이 악보의 마디를 기본 단위로 하는 마디 임베딩을 새롭게 구현하였다.

첫째. 학습용 미디 파일을 텍스트로 변환시킬 때, 마디

로 구분되도록 전처리를 한다.

둘째. 텍스트로 변환된 학습용 데이터로부터 각각의 마디에 대해서 인덱스를 부여하고 마디로 구성된 단어 사전을 만든다.

셋째. 딥러닝 프레임워크 중 하나인 케라스에서 제공하는 임베딩 레이어를 이용해서 각 마디를 밀집 벡터로 변환한다.

3.2 평가 방법

우리는 2.2절에서 자연어 처리 분야에서 사용되는 4가지 정량적 평가 방법들에 대해서 살펴보았다. 그 중에서 CIDEr은 자동작곡에서 유사 곡을 평가하는데 적합하지 않다고 판단이 되었다. 그 이유는 언어에는 불용어가 있지만 음악을 구성하고 있는 음표에는 불용어의 개념이 없이 하나 하나가 다 중요하고 의미를 가지고 있기 때문이다. 따라서 본 연구에서는 CIDEr를 제외한 나머지 3가지 정량적 평가 방법을 사용하였다.

3가지 정량적 평가 방법을 적용한 방법은 다음과 같다. 일반적으로 BLEU의 경우 기본 값이 4-gram까지 구해서 기하평균을 구한 다음 해당 결과를 BLEU 점수로 사용하지만 본 연구에서는 기하 평균을 사용하지 않고, 1-gram부터 4-gram까지 각각의 점수를 구해서 곡 평가에 사용한다. ROUGE의 경우에는 3-gram까지 단어 수준에서의 겹침을 평가함과 동시에 문장 수준에서의 유사도를 평가하기 위해서 ROUGE-L을 동시에 사용하기로 한다. METEOR의 경우에는 해당 평가 방법의 특성상 단어 수준과 문장 수준에서의 평가를 함께 구현하고 있으므로 그대로 사용하기로 한다.

4. 실험 결과

4.1 실험 데이터

미디 파일 1,034곡으로 구성된 노팅햄 데이터 세트를 이용하여 Table 1과 같이 학습용과 평가용으로 나누어 사용하였다. 데이터를 학습용과 평가용으로 분리하는 이유는 생성된 곡에 대한 정량적 평가에서 모방(copy)에 의한 고득점을 방지하기 위해서다. 학습용 데이터를 그대로 평가용으로 사용한다면, 자동 작곡 모델이 학습용 데이터를 그대로 모방해서 곡을 생성할 경우 생성된 곡의 정량적 평가 점수는 어떤 평가 방법을 사용해도 1에 가까운 값이 나올 것이고, 해당 모델은 사람이 작곡한 곡과 유사하게 곡을 생성한다고 해석을 하게 될 것이다. 우리의 목표는 인간이 작곡한 곡과 유사한 곡을 생성하는 것이지 모방이 아니다. 따라서 모방에 의한 고득점을 막기 위해서 인간이 작곡하되 학습에 사용되지 않은 데이터와 유사도를 평가하도록 설계하였다.

4.2 실험 방법

1) 실험 모델

Fig. 2에 있는 7가지 자동작곡 모델을 사용하였다. 원-핫

Table 1. Dataset Organization

data set(1,034 midi files)	
training data set	test data set
828(80%)	206(20%)

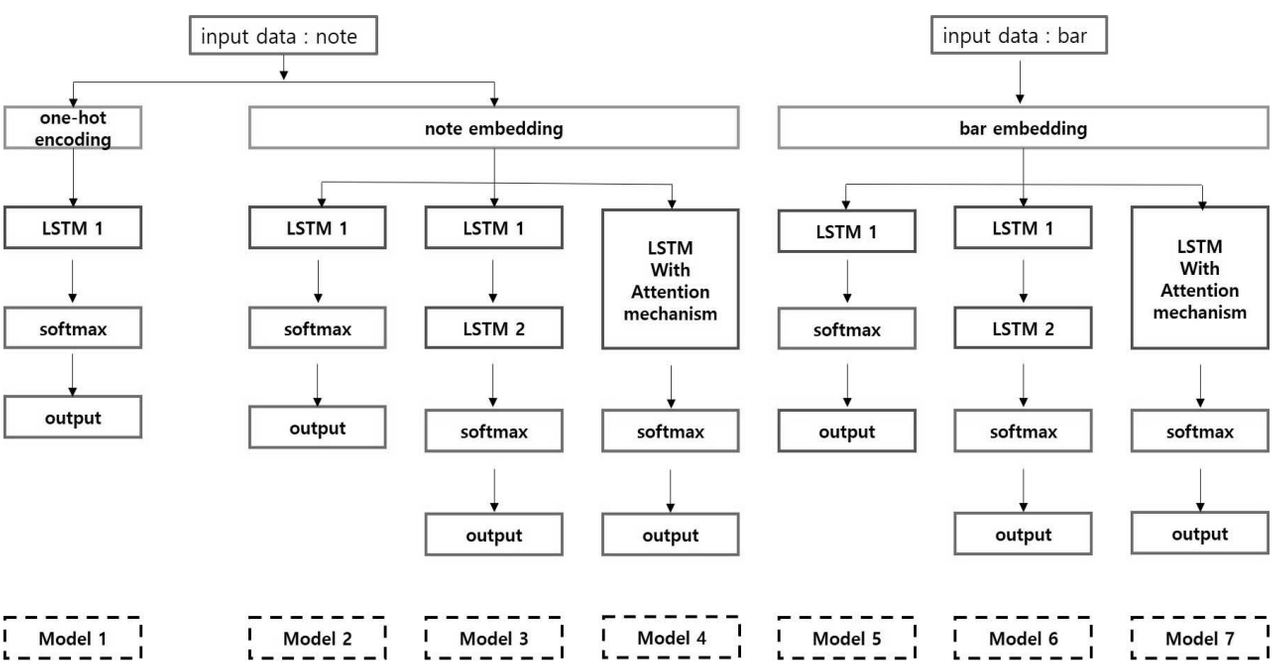


Fig. 2. Automatic Composition Models

인코딩 모델의 경우 원-핫 인코딩 방식의 특성상 희소벡터를 생성하기 때문에 학습을 진행할 때 컴퓨터의 메모리 문제를 발생시킨다. 그러므로 본 실험에서 원-핫 인코딩 방식과 음표 임베딩 방식의 차이를 비교 평가하는 실험은 구조가 가장 단순한 LSTM[16,17] 레이어 1개를 사용하는 순환신경망(recurrent neural network)[16-20] 모델을 이용하여 진행하고, 음표 임베딩 방식과 마디 임베딩 방식의 비교 평가는 LSTM 레이어 1개 및 2개를 사용하는 순환신경망 모델과 어텐션 메커니즘(attention mechanism)[21-24]이 구현된 순환신경망 모델을 각각 이용하여 진행한다.

2) 학습 조건

학습 조건은 다음과 같다.

a) 임베딩 레이어의 출력 크기 : 100차원
b) 미니 배치(mini batch)의 크기 : 512
c) 학습 횟수(epoch) : 300번으로 설정하되, 과대 적합을 방지하기 위하여 조기 종료 알고리즘을 사용하였다.
d) 조기 종료 조건 : 조기 종료의 기준(monitor)은 검증용 데이터 세트의 손실 함수 값으로 설정하였으며, patience는 10으로 설정하여 검증용 데이터의 손실 함수 값이 10회 연속으로 개선되지 않으면 종료하도록 설정하였다.
e) 손실 함수 : RMSprop
f) 학습률 : 0.002

4.3 평가 기준

우리는 인간이 작곡한 곡을 평가의 기준으로 설정해서 자동 작곡 모델별로 인간이 작곡한 곡과 얼마나 유사하게 곡을 생성하는지에 대한 비교 평가를 진행한다. 이를 위하여 2.2절과 3.3절에서 살펴본 자연어 처리 분야에서 사용되는 정량적인 평가 방법을 사용하였다.

4.4 실험 결과 및 분석

1) 곡 생성

Table 2와 같이 각 모델 별로 각각 50곡씩 생성하였으며, 곡의 생성 방법은 다음과 같다.

a) 평가용 데이터 세트(midi 파일 206개)에서 무작위로 50개 곡을 선정한다.
b) Table 2와 같이 음표 단위로 데이터를 입력하고 곡을 생성하는 model 1부터 model 4까지는 a)에서 무작위로 선정한 50곡으로부터 첫 번째 음표나 쉼표를 추출하여 입력 데이터로 사용하고, 마디 단위로 데이터를 입력하고 곡을 생성하는 model 5부터 model 7까지는 a)에서 무작위로 선정한 50곡으로부터 추출한 첫 번째 마디를 입력 데이터로 사용한다.
c) Table 2와 같이 각 모델별로 50곡씩 총 350곡을 생성한다.

Table 2. Experimental Models

One-hot encoding	Model 1	50 songs, 160 notes per song
Note embedding	Model 2	50 songs, 160 notes per song
	Model 3	50 songs, 160 notes per song
	Model 4	50 songs, 160 notes per song
Bar embedding	Model 5	50 songs, 44 bars per song
	Model 6	50 songs, 44 bars per song
	Model 7	50 songs, 44 bars per song

정량적 평가는 모델 별로 각각 생성된 50곡에 대한 평균값으로 진행한다.

2) 원-핫 인코딩과 음표 임베딩의 비교 평가

두 가지 방법에 대한 평가 결과는 Table 3과 같다. 음표 임베딩 방식이 원-핫 인코딩보다 어느 경우에나 더 높은 평가 점수를 얻어서 인간이 작곡한 곡과 유사한 곡을 생성한다는 것을 확인할 수 있었다. 두 가지 방식 모두 n이 증가할수록 n-gram BLEU 점수와 n-gram ROUGE 점수가 낮아지는데, 그 이유는 일반적으로 일치하는 연속된 음표의 수가 1에서 2, 3, 4로 늘어날수록 정확도(precision)와 재현율(recall)이 감소하므로 평가점수가 낮아지게 된다. n-gram BLEU 점수와 n-gram ROUGE 점수의 경우 음표 임베딩 방식이 원-핫 인코딩 방식보다는 상대적으로 높은 점수를 얻었으며, 곡 전체에서 배열의 순서가 일치하는 음표의 수(LCS)로 정확도와 재현율을 구하여 F-measure 점수를 구하는 ROUGE-L 점수도 음표 임베딩 방식이 상대적으로 높은 점수를 얻었고, 음표 단위의 유사도와 곡 전체 수준에서의 유사도를 모두 고려하는 METEOR 점수도 음표 임베딩 방식이 원-핫 인코딩 방식보다는 상대적으로 높은 점수를 얻었다. 우리가 3.1절에서 살펴본 것처럼 원-핫 인코딩의 결과 생성되는 원-핫 벡터는 희소 벡터로서 음표 간에 어떤 관계도 생성되지 않는다. 반면에 음표 임베딩 방식은 각 음표에 관계성을

Table 3. Comparison Evaluation Results for One-hot Encoding and Note Embedding

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Model1	0.7410	0.4105	0.1520	0.0129	0.2427
Model2	0.9815	0.5947	0.1675	0.0224	0.3276
	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L	
Model1	0.5154	0.2637	0.1356	0.4317	
Model2	0.5880	0.4162	0.2855	0.4686	

부여하고 음표 사이의 연관성이나 문맥을 활용할 수 있다. 이러한 차이로 인해서 새로운 곡을 생성할 때 음표 임베딩 방식이 원-핫 인코딩 방식보다 인간이 작곡한 곡과 더 유사한 곡을 생성한다.

Fig. 3의 (a)와 (b)는 두 가지 방식으로 전처리를 한 후에 순환신경망 모델을 통해 실제로 생성된 곡의 일부이다. 생성된 음표들을 살펴보면, (a)는 원-핫 인코딩 방식을 이용해서 생성된 것으로서 빨간색 네모로 표시한 부분처럼 특정 음표가 반복되어 생성되는 것을 확인 할 수 있었다. (a)의 경우 동일한 음의 지속적인 반복으로 인해서 곡 전체적으로 음의 구성이 단조로우며 음과 음 사이의 연결이 조화롭다고 생각하기 어렵다. 이와 달리 (b)의 경우 특정 음의 지속적인 반복 현상이 나타나지 않았고, 앞에서 서술한 바와 같이 정량적 평가 점수도 (a)보다 높게 나타났다. 이것은 음표 임베딩 방식이 원-핫 인코딩 방식보다 생성된 음의 연결이 인간이 작곡한 곡과 유사하다는 것을 의미하며 이는 앞에서 서술한 바와 같이 음표 임베딩 방식이 각 음표에 관계성을 부여하고 음표사이의 연관성이나 문맥을 활용할 수 있다는 것을 의미한다.

3) 음표 임베딩과 마디 임베딩의 비교 평가

두 가지 방법에 대한 평가 결과는 Table 4와 같으며, 동일한 조건을 가진 동일한 모델에서는 마디 임베딩 방식이 음표 임베딩 방식보다 더 높은 평가 점수를 얻어서 인간이 작곡한 곡과 유사한 곡을 생성한다는 것을 확인할 수 있었다. 그 이유는 다음과 같다.

우리가 3.1절에서 살펴본 것처럼 임베딩은 음표나 마디와

Table 4. Comparison Evaluation Results for Note Embedding and Bar Embedding

	BLEU -1	BLEU -2	BLEU -3	BLEU -4	METEOR
Model2	0.9815	0.5947	0.1675	0.0224	0.3276
Model5	0.9921	0.8626	0.5762	0.3062	0.4519
	ROUGE -1	ROUGE -2	ROUGE -3	ROUGE -L	
Model2	0.5880	0.4162	0.2855	0.4686	
Model5	0.7322	0.5700	0.4286	0.5396	
	BLEU -1	BLEU -2	BLEU -3	BLEU -4	METEOR
Model3	0.9750	0.5970	0.1835	0.0953	0.3133
Model6	0.9889	0.8545	0.5625	0.3006	0.4499
	ROUGE -1	ROUGE -2	ROUGE -3	ROUGE -L	
Model3	0.5845	0.4106	0.2669	0.4710	
Model6	0.7263	0.5642	0.4186	0.5354	
	BLEU -1	BLEU -2	BLEU -3	BLEU -4	METEOR
Model4	0.9599	0.6848	0.2832	0.0490	0.3181
Model7	0.9902	0.8597	0.5612	0.2911	0.4454
	ROUGE -1	ROUGE -2	ROUGE -3	ROUGE -L	
Model4	0.5347	0.3582	0.2515	0.4535	
Model7	0.7275	0.5679	0.4258	0.5340	

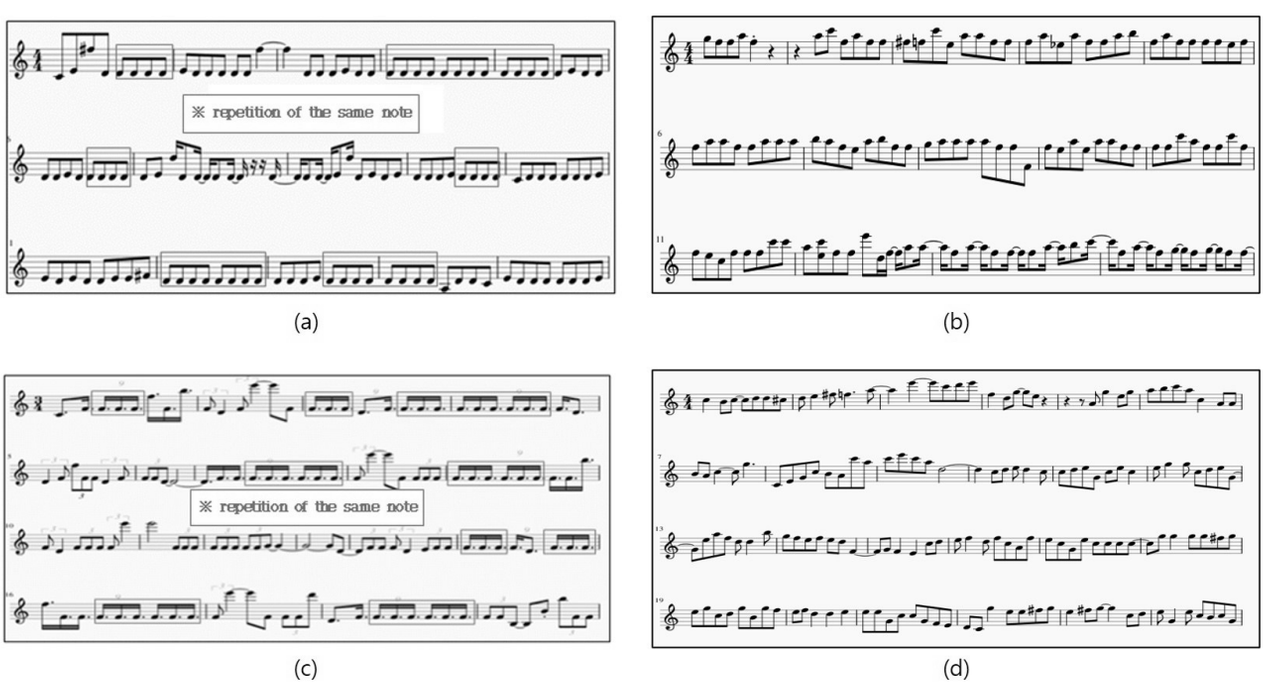


Fig. 3. Examples of Creating a Song : (a) Part of a Song Created with One-hot Encoding Method, (b), (c) Part of a Song Created with Note Embedding Method, and (d) Part of a Song Created with Bar Embedding Method

같은 정보를 쓰임새와 관계성을 감안하여 벡터화 한다. 그러므로 LSTM 레이어가 음표와 음표 또는 마디와 마디 사이의 관계를 학습하는 것이 원-핫 인코딩보다 더 수월하게 된다. 즉 음표 임베딩은 음표 수준에서 마디 임베딩은 마디 수준에서 유사한 쓰임새와 관계에 있는 것은 유사한 벡터가 되기 때문에 LSTM에서 곡을 학습할 때 학습이 원활하게 되고 학습이 정확하지 않아도 유사한 음표와 마디로 출력하게 된다.

학습에 사용한 미디 파일 828곡에 들어있는 음표의 총 수는 156,405개이고 음표의 종류는 384개, 마디의 종류는 9,149개이다. 음표의 종류는 비교적 작기 때문에 임베딩 시에 쓰임새나 관계까지 학습해서 벡터화 하는 것이 어렵다. 그에 비하여 마디의 종류는 음표의 종류의 대략 30배로 마디의 쓰임새나 관계를 학습하기에 충분하다. 그러므로 음표 임베딩에 비하여 마디 임베딩이 더 유리하다.

Fig. 3의 (c)와 (d)는 두 가지 방식으로 전처리를 한 후에 순환신경망 모델을 통해 실제로 생성된 곡의 일부이다. 생성된 음표들을 살펴보면, 원-핫 인코딩 방식보다는 적지만 음표 임베딩 방식의 경우도 동일한 음을 반복해서 생성하는 것을 확인 할 수 있었다. 이는 4.4절에서 살펴본 것처럼 음표 임베딩 방식이 마디 임베딩 방식보다 정량적 평가 점수가 더 낮은 이유가 되며, 음표 임베딩 방식의 학습 능력이 마디 임베딩 방식보다는 떨어진다는 것을 의미한다.

5. 결 론

우리는 본 논문에서 자동작곡이 보다 더 작곡가가 작곡한 곡과 유사한 곡을 생성하게 하기 위하여 음표 임베딩 방법과 마디 임베딩 방법을 제안하였다. 곡의 유사도를 정량적으로 비교하기 위하여 자연어 처리 분야에서 사용되는 정량적 평가 방법 BLEU, ROUGE, METEOR를 이용하였다. 정량적 평가결과 음표 임베딩 방식이나 마디 임베딩 방식이 원-핫 인코딩 방식 보다 인간이 작곡한 곡과 더 유사하게 곡을 생성한다는 것을 확인하였다. 이를 통하여 본 논문에서 제안한 음표 임베딩 및 마디 임베딩이 자동작곡에서 보다 더 작곡가가 작곡한 곡과 유사한 곡을 생성할 수 있음을 보였다. 특히 마디 임베딩 방식이 음표 임베딩보다 더 좋은 결과를 보였는데 이는 마디의 종류가 음표의 종류보다 풍부해서 보다 더 쓰임새와 관계를 잘 임베딩하여 벡터화한 결과로 보인다. 정량적 평가방법으로 자연어처리에서 사용하는 평가방법을 사용하였으나 곡의 평가와 자연어처리 평가에는 차이점이 존재한다. 이를 극복하기 위하여 향후 정량적으로 곡을 평가하는 방법을 연구하는 것이 필요하다.

References

[1] H. Chu, R. Urtasun, and S. Fidler, "Song from PI: A musically plausible network for pop music generation" *arXiv preprint arXiv:1611.03477*, 2016.

[2] F. Shah, T. Naik and N. Vyas, "LSTM Based Music Generation," *2019 International Conference on Machine Learning and Data Engineering*, 2019.

[3] P. Chen and E. Xu , "CS 224 N project report: From Note 2 Vec to Chord 2 Vec," 2019.

[4] S. R. Hwang and Y. C. Park, "Chord-based stepwise Korean Trot music generation technique using RNN-GAN," *The Journal of the Acoustical Society of Korea*, Vol.39, No.6, pp. 622-628.

[5] KakaoBrain. Similarity Method Between Words [Internet], https://www.kakaobrain.com/blog/6.

[6] L. C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Computing and Applications*, Vol.32, No.9, pp.4773-4784, 2020.

[7] B. Logan, D. P. Ellis, and A. Berenzweig, "Toward evaluation techniques for music similarity," *The MIR/MDL Evaluation Project White Paper Collection*, Vol.3, pp.81-85, 2003.

[8] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp.311-318, 2002.

[9] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of the Workshop on Text Summarization Branches Out*, pp.74-81, 2004.

[10] S. Banerjee and A. Lavie "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp.65-72, 2005.

[11] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.4566-4575, 2015.

[12] S. Sharma, L. E. Asri, H. Schulz, and J. Zumer, "Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation," *arXiv preprint arXiv: 1706.09799*, 2017.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proceedings of Workshop at ICLR*, 2013.

[14] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1532-1543, 2014.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, pp.427-431, 2017.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol.9, No.8, pp.1735-1780, 1997.
[17] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp.338-342, 2014.
[18] T. Mikolov, M. Karafiat, L. Burget, J. Cernock, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pp.1045-1048, 2010.
[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv: 1409.3215*, 2014.
[20] R. Jozefowicz, W. Zaremba, and B. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning*, pp.2342-2350, 2015.
[21] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2016.
[22] K. Xu, J. Ba, R. Kiros, K. H. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "How, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, pp.2048-2057, 2015.
[23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, pp.1480-1489, 2016.
[24] Z. Lin, M. Feng, S. N. Santos, M. Yu, B. Xiangl, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

이 영 배

https://orcid.org/0000-0002-0639-9759
e-mail : newton89@naver.com
2003년 한국방송통신대학교 경영학과 (학사)
2021년 한성대학교 지식서비스&컨설팅대학원 미래융합컨설팅학과(석사)
2021년 ~ 현 재 ㈜보인정보기술 개발2팀 차장
관심분야 : Natural Language Processing, Computer Vision

정 성 훈

https://orcid.org/0000-0002-9674-4543
e-mail : shjung@hansung.ac.kr
1988년 한양대학교 전자공학과(학사)
1991년 KAIST 전기및전자공학과(석사)
1995년 KAIST 전기및전자공학과(박사)
1996년 ~ 현 재 한성대학교 기계전자공학부 교수
관심분야 : Artificial Intelligence, Systems Biology, Fusion Engineering