A Method for Calculating Melody Concatenation Cost based on BiLSTM.

Tatsunori Hirai※, Shun Sawada※※

Abstract

In this paper, a model to calculate the cost of concatenating melody fragments using BiLSTM is proposed. The cost calculated using the proposed model quantifies the manner in which two melody fragments (measures) can be connected naturally. The proposed concatenation cost is based on the likelihood of note transitions at the connection boundary and a BiLSTM model that is trained by shuffling existing melodies by measures and determining their connection points. We compared the accuracy of determining the connection points of melody fragments against several network configurations and explored suitable models for calculating concatenation costs. The comparison showed that the highest accuracy was obtained with the proposed BiLSTM model. In addition to proposing a model for calculating concatenation costs, this study discusses how concatenation costs can be used to support music creation.

Key words: Melody processing; LSTM; RNN; melody concatenation cost

Introduction

When composing songs, it is necessary to create a complete melody, typically a melody for the verse and for the chorus. However, during the creation process, composers often come up with short melody phrases that are not sufficient for a complete verse or chorus. Moreover, it is relatively easy, even for people who are not experts in music composition, to come up with a short melody phrase, for example while humming. Conversely, creating a melody for a complete song is not easy. This suggests that the most difficult part of the song creation process is connecting short melody fragments such that their combination comprises a whole song. Herein, we propose the cost of concatenating melodies as a measure of the naturalness of the connections between melody fragments.

If we can create longer melodies by joining short melody fragments, we believe that we can associate the act of casually humming to more serious music production. In addition, if we can determine which melody fragments can be connected naturally, creating a medley or mashup music will be much easier. Thus, with the aim of supporting music production, this study investigates the concatenation cost of melody fragments.

In general, information processing systems that deal with musical melodies face difficulties in performing evaluations. To avoid subjective evaluations, some music generation studies have relied on the prediction accuracy of subsequent melodies in the training data or measuring the estimation accuracy of the missing parts. Such training methods allow us to build a model that deals with quantitative accuracy. Consequently, even in the context of music generation, where it is difficult to evaluate results, we can infer, to some extent, whether the system is good or bad. Conversely, for a system that directly evaluates the quality of the generated music via subjective evaluation experiments, it is difficult to assess the system quality beyond the results provided for particular subjective experiments written in the paper. Therefore, when a new method

※ Komazawa University Author contact: thirai@komazawa-u.ac.j

※※ Tokyo University of Science Author contact: sawada@rs.tus.ac.jp

is proposed, it is difficult to compare the proposed system with previous systems under the same conditions.

Therefore, this study proposes a model that can evaluate the quality of the system as objectively as possible by incorporating a quantitative evaluation measure, i.e., the melody concatenation cost, into the learning task. Specifically, we shuffle the melodies in the training data by measure, input the shuffled and unshuffled melodies into the model, and build the model based on learning whether or not the melodies are shuffled. This allows us to construct a model that can determine whether a melody is originally connected or unnaturally connected and also to quantitatively measure the accuracy of the determination. In addition to the naturalness of the connection between measures, we consider the transition probability of notes at the boundary between measures because the connection between notes at the connection point is not always natural if we only consider the connection between measures of the melody. Using these methods, we propose a concatenation cost that considers the naturalness of the connection per measure as well as the naturalness of the transition per note at the connection boundary.

Related Work

To generate new melodies by reusing existing melodies using the concatenation cost of melodies, Bretan et al. proposed a melody unit-selection method using deep learning models[1] . Unit selection is a generation method that has been actively studied in the text-to-speech field. In the unit selection method proposed by Bretan et al., melody generation is performed by searching the database for melodies that follow the input melody by considering semantic relevance and concatenation cost. First, to determine the semantic relevance, an embedding vector of 500 dimensions is obtained using a two-layer autoencoder for the features extracted using the bag-of-words approach. The semantic relevance of each unit can be determined by constructing an LSTM model of the transitions of the embedding vectors. Next, the top 5% of melody units in the database that exhibit high semantic relevance to the input melody are identified. The semantic relevance and note-level concatenation cost for the melody units are calculated, and subsequent melody units based on the combination of the two measures will be selected. The note-level concatenation cost is obtained by training a model that predicts the next note using a multilayer LSTM. In summary, this method first narrows down the dataset to the top 5% of melodies that are semantically close to the input melody and then selects the next melody unit based on semantic closeness and note-by-note concatenation cost to generate a longer melody. Because the objective of our method is to calculate the concatenation cost between melodic fragments rather than to generate melodies, we investigate whether the part of Bretan et al.'s method that determines the semantic relationship between melody units can be replaced by the concatenation cost. In addition to the note-by-note concatenation costs, we consider concatenation costs over longer time spans (i.e., measures).

Various studies have explored methods to create new music by combining existing melodies. Cope[2] proposed a method to create new music by segmenting existing melodies into small pieces and labeling each piece based on its characteristics to reuse and recombine them. Unlike our proposed method and the method proposed by Bretan et al., Cope's approach is to reconstruct melodies based on rules without relying on machine learning-based modeling. Kitahara et al. proposed JamSketch, which uses a genetic algorithm to generate improvised melodies in real time based on the approximate shape of the melody input by the user[3] . They used existing melodies from a dataset in their genetic algorithm-based generation. Although JamSketch

1 Bretan, M., Weinberg, G., amd Heck, L.: A Unit Selection Methodology for Music Generation using Deep Neural Networks, Proceedings of the International Conference on Computational Creativity 2017 (2017).

2 Cope, D.: One Approach to Musical Intelligence, IEEE Intelligent Systems and their Applications, Vol. 14, No. 3, pp. 21–25 (1999).

3 Kitahara, T., Giraldo, S., and Ramírez, R.: JamSketch: Improvisation Support System with GA-Based Melody Creation from User’s Drawing, Proceedings of the International Symposium on Computer Music Multidisciplinary Research, pp.509–521 (2017).

does not use existing melodies directly, it can be considered an example of using existing melodies for melody generation. The Continuator, proposed by Pachet, is another system that generates a new melody based on an existing melody[4] . The Continuator is an interactive system that divides a melody into small fragments and models the transitions from one fragment to another using a tree-structured Markov chain to search the training data for a suitable melody to follow the input melody.

Generating new content by reusing existing content has been attempted in various domains. For example, in the image synthesis method PatchMatch, which is an image interpolation process, small patch areas that correspond to the interpolated area are searched for in the same image and combined to create new interpolated image content[5] . Hirai et al. proposed a method to generate music videos automatically, wherein new music videos are generated by searching for video fragments that match the music from existing video databases and connecting the found fragments[6] .

It is evident that the development of methods to generate content by reusing existing content is an active research area. Herein, we investigate a method to calculate the concatenation cost between melody fragments. We believe that our proposed approach will be an elemental technology to realize the generation of new melodies by reusing existing melodies.

Data Preparation

When constructing a deep learning model for creating melodies, deciding the type of database to use, and determining the manner of representing the melody data are crucial because the accuracy and the results can vary considerably depending on these selections. Because the database type and data representation technique are also related to the reproducibility of the method, they are described in detail in this section.

Dataset

Herein, we use the melody data extracted from the Lakh MIDI dataset[7] . This dataset comprises 176,581 MIDI files collected from the Web. Melody files were extracted from this dataset according to the preprocessing method used by Hirai et al. to learn distributed representations of melodies[8] . We obtained 10,853 melodies and used 10,736 melodies with length greater than two measures (i.e., 32 sixteenth notes). The 117 excluded melodies are mainly empty tracks due to the unmaintained MIDI data from the web. As the next step, the preprocessing method proposed by Hirai et al. was used to estimate the tones of the acquired melodies and transpose them as necessary to ensure that all melodies in the dataset were in the same key.

The Lakh MIDI dataset includes many MIDI melody files that do not include the annotated symbolic information owing to the addition of performance expressions. In addition, many of these files contain noisy data due to inappropriate labeling. Although cleaner datasets are available, the number of songs in such datasets is limited. We selected the large Lakh MIDI because our priority is to conduct training using as much data as possible.

---

4 Pachet, F.; The continuator: Musical interaction with style, Journal of New Music Research, Vol. 32, No. 3, pp.333–341 (2003).

5 Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B.: PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, ACM Transactions on Graphics, Vol. 28, No. 3 (2009).

6 Hirai, T., Ohya, H., and Morishima, S.: Automatic Mash up Music Video Generation System by Perceptual Synchronization of Music and Video Features, Proceedings of the ACM SIGGRAPH 2012, pp.449:1 (2012).

7 Raffel, C.: Learning-based Methods for Comparing Sequences, with Applications to Audio-to-midi Alignment and Matching, PhD Thesis, Columbia University (2016).

8 Hirai, T. and Sawada, S: Melody2Vec Distributed Representations of Melodic Phrases based on Melody Segmentation, Journal of Information Processing, Vol. 27, pp.278–286 (2019).

Data Representation

A melody is composed of a sequence of notes comprising a combination of pitch and note values. MIDI data also contain other information, such as velocity, which represents the intensity of the performance. However, only the note sequences are processed herein. When the melody is extracted from the MIDI data, a note sequence is represented by an MIDI note number and a tick (the length of a quarter note). The extracted and transposed melody is processed as a sequence of notes in a text format. Our text representation of a melody is simpler than that of an ABC notation, which only includes tick and note sequence information.

To calculate the concatenation cost, note sequences must be represented in a form that can be input to the model. The data-representation method must consider the strategy employed to calculate the concatenation cost of the melody fragments. In the concatenation cost proposed herein, we focus on both the validity of the connection at the measure-level and the note-level and construct a model that integrates them.

We assume that a measure is four-quarter-notes long (four quarters of a beat) and segment the melody data by four quarter notes. Moreover, a one-measure note sequence is divided into sixteenth notes and represented by a one-hot vector containing only the note names. Such a representation of measures with sixteenth notes as the smallest unit is a common method used in methods such as DeepBach by Hadjeres et al.[9] . The one-hot vector is a 13-dimensional vector containing the names of the 12 notes and rests; as there are 16 of these one-hot vector per measure, it is a 13×16 matrix. Here, when a measure is divided into sixteenth-note units, information about the note boundaries is lost when notes with the same name are played in succession. To prevent this, a state called "hold" is often introduced, which indicates the state in which the previously played sound is continued. However, in the case of the proposed method, when experiments were conducted with and without "hold," learning was more successful when hold was not introduced. Therefore, we adopted the 13-dimensional one-hot vector representation for developing the measure-level concatenation validity evaluation model. We only use the note names and ignore the octave information to prevent the learning from failing owing to the diversity in the states of input data caused by the variation of octaves.

However, when evaluating the validity of note-level concatenation, it is important to include the octave information and note length in the data representation. Therefore, for note-level concatenation, we use the octave information and note length (pitch + note value) as well as the note names. Octave information is included to evaluate the validity based on actual examples of note transitions. To handle the variety of notes that appear in a melody, our data representation approach considers the melody as a sentence, and the notes are converted into words. Furthermore, we assign an ID to each note in a melody in the dataset. We represent the data such that the more number of candidate note-transition patterns included in the dataset, the more valid is the transition.

Melody Concatenation Cost Model

As mentioned in the previous section, we consider the validity of the measure-level connections and note-level concatenation when calculating the concatenation cost of a melody. In practice, it would be possible to consider the validity of connections at any resolution, i.e., not limited to measures and notes; however, we leave the extension to be included in future work. In this section, we discuss the validity of measure-level and note-level concatenations as well as the methods for integrating them.

9 Hadjeres, G., Pachet, F., and Nielsen, F.: DeepBach: a Steerable Model for Bach Chorales Generation, Proceedings of the 34th International Conference on Machine Learning, pp.77:1362–1371 (2017).

Measure-level Concatenation Score

We construct a model to evaluate the validity of the measure-level concatenation using the measure-level one-hot vector as the input. We construct a deep learning model that can determine whether two melodies of any two input measures are connected to each other.

When building a model for melodies, it is desirable to develop a model that can consider time-series information. Toward this end, we implement several types of RNN models, compare their accuracy, and select the highest accuracy model that is to be adopted.

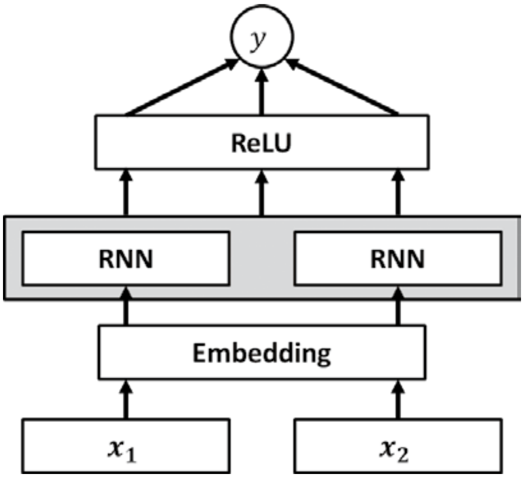

Figure 1 Network framework to determine measure-level concatenation cost

The network diagram of the RNN-based model constructed herein is shown in Figure 1. We compared several types of RNN models; depending on the model, the RNN layer shown in Figure 1 is replaced by LSTM, 2-layer LSTM, and BiLSTM. The data input to the model is a 13-dimensional one-hot vector (13×16 matrix data) of 12 notes and rests, represented in units of 16th notes, for two measures (13×32). The data flow in the framework of the RNN model constructed in this study is as follows.

1. Input the first measure of data x_1 to the embedding layer and then to the RNN layer to obtain the output h_1 of the last hidden layer.
2. Embed the second measure of data x_2 and then input to the RNN layer to obtain the output x_2 of the last hidden layer.
3. Combine h_1 and h_2 and input it to the fully connected layers (2 or 3 layers) to obtain the binary classification output $[0,1]$.
4. Train the network using the auto-annotated training labels, 1 if the two measures that have been input are a combination of measures that were actually connected, and label 0 otherwise.

In the above framework, we trained an RNN model to make decisions about the concatenation between the measures for melodies in the dataset introduced in previous section. We also experimented to see how the accuracy changes when we change the RNN layer in Figure 1 from a simple RNN to LSTM, 2-layer LSTM, and BiLSTM. For training, we split the 10,736 melodies in the dataset into training and test data at a ratio of 9:1. If we use the training data as is, all data will be positive examples, i.e., examples in which the consecutive measures are originally connected. Therefore, for each melody in the dataset, we prepared a positive example in which the original melody was left unchanged and a negative example in which every even-numbered measure was replaced with a random melody (i.e., the consecutive measures are not connected). This increased the number of available data by a factor of two. We did not use a completely random melody. Every even-numbered measure was replaced with a different measure from a different melody in the dataset. In all the compared RNN models, the dimensionality of embedding was set to 25 and the dimensionality of the output of the hidden layer of the RNN was set to 50. Only in the case of BiLSTM, we increased the number of the fully connected layer to two layers because the dimensionality of the output from the hidden layer is doubled in BiLSTM. In all models, the loss function was binary cross-entropy, and training was performed until early stopping became effective. The RNN, LSTM, 2-layer LSTM, and BiLSTM models were trained for 459, 533, 459, and 362 epochs, respectively.

When the melody data of any two measures are input to a model constructed above, the validity of the concatenation between the measures expressed on a scale of [0,1] can be determined. Herein, we denote this as the measure-level concatenation score S_m.

Note-level Concatenation Score

Although the measure-level concatenation score described in previous section can consider the naturalness of connections at the granularity of measures, it cannot consider the smoothness of note connections at measure boundaries. The model that calculates the concatenation score at the measure level only deals with the changes of note names within a measure, and the information of actual notes is lost. Therefore, the connection of notes must be considered in relation to the measure-level score. Therefore, we introduce a measure to evaluate the validity of note concatenation at measure boundaries. Specifically, based on what the last note of the previous measure and the first note of the following measure are when all the melodies in the training data are divided into measures, we record what type of transition is a natural connection between the notes across the measures. Based on the ID representation of notes, we count the number of transitions from any ID to any ID at the measure boundary and divide the counted value by the most frequent value of the transition for each type of note to obtain a value on a [0,1] scale, corresponding to the validity of the transition. We refer to this as the note-level concatenation score S_n. This score corresponds to the transition probability of a note; however, the scale is normalized to the maximum value, i.e., the score value is 1 for the most likely transition. In the case of transitions between notes that are not included in the training data, the score is 0, which makes it difficult for connections between notes that are not available in the training data.

Melody Concatenation Cost

The concatenation cost of the melody was calculated by combining the measure-level concatenation score S_m and the note-level concatenation score S_n. Both S_m and S_n were represented on a scale of [0,1], with higher scores indicating more valid connections. Using these values, we defined the melody concatenation cost C as follows:

$$C = 1 - \{\alpha S_m + (1-\alpha) S_n\} \quad (1)$$

Here, α is a weight factor of [0,1], where a larger value emphasizes measure-level connections and a smaller value emphasizes note-level connections. Herein, for the experiments described below, we use α=0.75 to emphasize the measure-level concatenations.

Results

In this section, we describe the accuracy of the proposed concatenation cost in determining the connection points of melodies. Specifically, we compare the RNN models and the obtained results of the calculation of the concatenation cost for actual melody pairs.

Comparison of Melody Concatenation Detection Accuracy

To evaluate the melody connection cost calculated using the method described in previous section, we calculate and compare the determination accuracy (correct answer rate) of the melody connection point, i.e., determining whether the input melody of two measures is connected. In this section, we first compare RNN models for calculating measure-level concatenation scores and then compare the determination accuracy of the measure-level concatenation scores, note-level concatenation scores, and the proposed concatenation costs that considers both these scores.

Comparison within RNN model

First, to determine which RNN model should be adopted when calculating measure-level concatenation scores, we compared the accuracy of determining the melody connection points when using each RNN model. Here, 90% of the dataset (i.e., 90% of the total number of songs, not the number of measures) was used to train the model. The remaining 10% of the dataset was used as a test data for the model described in previous section to determine whether the two measures of a melody were originally connected or whether another melody was connected. Table 1 shows the determination accuracy when using a simple RNN, LSTM, two-layer LSTM, and BiLSTM as the RNN layer. The accuracy was determined by thresholding the scores output by each model in the range [0,1] at 0.5 and considering that the measures were originally connected if they were above the threshold.

Table 1 Comparison of the accuracy by measure-level score.

RNN model	Accuracy [%]
RNN	82.82
LSTM	84.08
2layer-LSTM	84.34
BiLSTM	**84.38**

As shown in Table 1, by adopting BiLSTM as the RNN layer, we determined whether the measures were originally connected or randomly connected with the highest accuracy of 84.38%. The accuracy may not always reach 100% because some random connections may be reasonable measure-to-measure connections. In other words, some of the negative examples may contain measure combinations that are accidentally connected in a natural way. Because the accuracy of the random prediction was approximately 50%, it can be said that the measure-level concatenation score determined using our method can be considered a reasonable measure.

The decision accuracy may be improved by examining all network components to the better module for accuracy improvement and exploring optimization methods, such as introducing a transformer. In future, when we search for a more accurate model, we can objectively compare models by conducting experiments similar to this one.

Comparison of Melody Concatenation Determination Accuracy

Next, we calculated and compared the determination accuracy based on the note-level concatenation score and the determination accuracy based on the concatenation cost derived using Equation (1), which considers both. Similar to the measure-level scores, for the note-level concatenation scores, we used 90% of the dataset to calculate note transitions and 10% for testing. To determine connections based on note-level scores, a natural connection was determined if the concatenation score between two notes at the measure boundary was 0.1 or higher. The threshold was set to 0.1 because the estimation accuracy of the threshold value around 0.1 was high.

Herein, we calculated the accuracy for 10% of the test data. Here, we set the threshold value to 0.5 and determined that a connection is natural if the concatenation cost is less than the threshold value. The BiLSTM model was used for calculating the measure-level concatenation score, and the weight factor α for calculating the cost was set to 0.75.

Table 2 Comparison of the accuracy by each score/cost.

model	accuracy [%]
measure-level concatenation cost	**84.38**
note-level concatenation cost	77.99
concatenation cost	84.13

The accuracies obtained for determining the connection points based on the measure-level concatenation score, note-level concatenation score, and concatenation cost are shown in Table 2.

As shown in Table 2, from the perspective of purely determining the connection between measures, the accuracy was the highest when using the measure-level concatenation score. This is a natural result because the measure-level concatenation scores are modeled by learning to minimize the loss in making connection decisions. When considering note-level concatenation scores and introducing concatenation costs, the accuracy of the melody-to-melody connection evaluation is reduced; however, the validity of note-level connections can be considered, thus, it is not necessarily ineffective. Because the naturalness of the transition between the last note of the previous measure and the first note of the subsequent measure can be considered, introducing the proposed concatenation cost has an advantage, i.e., the local naturalness of measure boundaries can be considered. In addition, by adjusting the weight value a, the user can change the importance of note connection and measure connection in the concatenation cost calculation. In the future, we plan to implement an interface to allow users to adjust the weight value.

In this comparative evaluation, each score and cost were evaluated via threshold processing. Therefore, changing the threshold and parameters values can easily change the results. Moreover, if we set the threshold of the connection determination at the note level to 0.5, the accuracy is reduced to 67.32%, and if we set the value of α to 0.5 in the concatenation cost calculation, the accuracy drops to 77.63%. Although the same data are used, the accuracy values are not completely comparable under the same conditions.

Calculation of the Concatenation Cost

Using the proposed model, we can calculate the concatenation cost between some melody fragments. We calculated the concatenation costs for the five combinations of

Figure 2 Five measures used in the concatenation cost calculation experiment

Table 3 Calculation of the concatenation cost of existing melodies.

		x_2				
		m_1	m_2	m_3	m_4	m_5
x_1	m_1	0.56	0.62	0.60	0.70	0.86
	m_2	0.70	0.58	0.65	0.74	0.82
	m_3	0.71	0.74	0.54	0.43	0.80
	m_4	0.67	0.75	0.56	0.59	0.94
	m_5	0.90	0.92	0.75	0.67	0.68

measures m_1 to m_5 , as shown in Figure 2. The results are listed in Table 3.

In the measures shown in Figure 2, m_1 and m_2 pair, and also m_3 and m_4 pair are originally connected measures and m_5 is a random one-measure melody prepared herein. In Table 3, the underlined values represent the concatenation costs between the originally connected measures (m_1 and m_2 pair, and, m_3 and m_4 pair). This result shows that the concatenation cost between originally connected measures is relatively low, and the concatenation cost of the combinations that are not originally connected is comparatively high. Similarly, the cost of the transition from m_4 to m_3, which is the reverse of the original connection, is observed to be low. However, the melody is considered to be naturally connected even in the reverse order. The concatenation cost for connecting identical measures is also lower, which may correspond to the frequent use of repetition in melodies.

In the case of connections between originally unconnected measures, the transition from m_1 to m_3 has a relatively low concatenation cost. When this combination was connected and listened to, the transition was natural within the scope of the author's subjective impression. For the transition from m_4 to m_5 , which had the highest concatenation cost, there was no sense of discomfort at the connection point because the connection felt like two completely different melodies caused by a rest at the end of m_4.

The relationship between the concatenation cost and auditory naturalness of the connection obtained using this method will be evaluated in the future.

Directions for Future use of Concatenation Costs

The concatenation cost between melody fragments can be applied to melody generation, as described in the unit-selection-based melody generation method proposed by Bretan et al.[10]. Based on an arbitrary input melody fragment, a longer series of melodies can be obtained by sequentially selecting melodies with the lowest concatenation cost. However, reusing of an existing melody might be considered as a different act from creating new original music. Therefore, in the future, we would like to explore the possibility of creating new music by reusing existing melodies. For example, a method that supports the creation of long melodies by creating a database composed of short melody fragments, such as those created by humming, and combining them. In addition, a system that automatically generates a large amount of short melody phrases and allows the user to interactively select the continuation of the created melody using our proposed concentration cost method could be developed.

In the future, we would like to realize an interface to support melody creation by applying the connection cost of melody elements proposed herein. We will also explore the possibility of composing short melodic phrase-based compositions. For example, we intend to study the feasibility of music production by a large number of composers by combining short melody pieces produced by dozens of users. Splice[11] and other sound libraries have released a large number of short sound materials that are being used as part of the work of creators globally. We believe that melody could be handled similar to sound materials, and we intend to explore this possibility in the future. The calculation of melody concatenation costs is a technique that will lead to the realization of such a future.

Conclusion

We proposed a BiLSTM-based model to calculate the concatenation cost between melody fragments. The concatenation cost was defined based on the naturalness of the connection per measure as well as the naturalness of the transition per note at the connection boundary. We compared the accuracy of several types of RNN models for determining whether a melody is originally connected. The results demonstrate that in terms of determining connection points, the BiLSTM model was more accurate. We also calculated and compared the accuracy of determining connection points using measure-level concatenation scores, note-level concatenation scores, and concatenation costs. The determining connection points of melodies, which was used as an evaluation measure herein, can be used as an objective index for evaluating the model of melody generation. We believe that the evaluation task adopted herein can be used without modification in future experiments aimed at building a more accurate model.

In the future, we will attempt to realize a melody creation support system that uses the concatenation cost of melody elements proposed in this study.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers JP19K20301.

10 Bretan, M., Weinberg, G., amd Heck, L.: A Unit Selection Methodology for Music Generation using Deep Neural Networks, Proceedings of the International Conference on Computational Creativity 2017 (2017).

11 https://splice.com/