Global Prediction of Time-span Tree by Fill-in-the-blank Task

Riku Takahashi[1],Risa Izu[1], Yoshinari Takegawa[1] and Keiji Hirata[1]

Future University Hakodate
g2122038@fun.ac.jp

Abstract. Time-span trees in A Generative Theory of Tonal Music (GTTM) have global and local relationships. However, no analysis based on global relationships has been done, and higher-order mechanisms have not been clarified. Therefore, in this research, we will clarify this mechanism by masking the time-span tree as it is and using it as a fill-in-the-blank task. To experiment with the fill-in-the-blank task, we vectorized and embedded the tree structure. We also extended the data by changing the pitch and masking. As a result of experiments, it is possible to predict higher layers when masking a small maximum time-span.

Keywords: Generative theory of tonal music (GTTM), time-span tree, LSTM, Seq2Seq, blockview, skip-thought

1 Introduction

In cognitive science, how humans listen to music is an important factor. When listening to music, people do actively listen while predicting the next melody, harmony, and rhythm. Emotional arousal during listening to music is said to be related to betrayal of the listener's conscious and unconscious predictions[1]. It has been reported that when an unpredictable change in pitch occurs while listening to a melody, people react psychologically and physiologically[2]. It has also been found that the memorability of a melody is related to the predictability [3]. For these reasons, we find that human perception of music is simultaneously analyzing and predicting. Therefore, music analyzers are required to be able to perform multiple analysis and prediction.

In addition, music theory shows that there are not only local relations but also global relations. A Generative Theory of Tonal Music (GTTM) [4], which is a cognitive music theory, describes local and global relationships, and is implemented on computers. In particular, GTTM's time-span tree implementation has been attempted [5, 6]. So far, however, there has been little research about how to implement higher-order mechanisms. Specifically, the current algorithm inputs scores in specific batches and analyzes the time-span tree based on local relationships. However, when humans listen to music, they analyze and make predictions at some point. We never listen to the whole music

and analyze it in batches. In fact, referring to Fig. 1, the subtree of C C and the subtree of G G are composed one subtree, so they have a local relationship. Similarly, there is a global relationship between C C and A A. On the other hand, the subtree of G G and the subtree of A A have not local relationship and global relationships. However, the subtree of C C, G G, and A A can be seen as three consecutive syllables of the same two sounds. Therefore, there is also a relationship between G G and A A as syllables. However, due to the reduced subtrees of the GTTM rules, it is difficult to find hidden global relations for subtrees of such syllables and phrases.

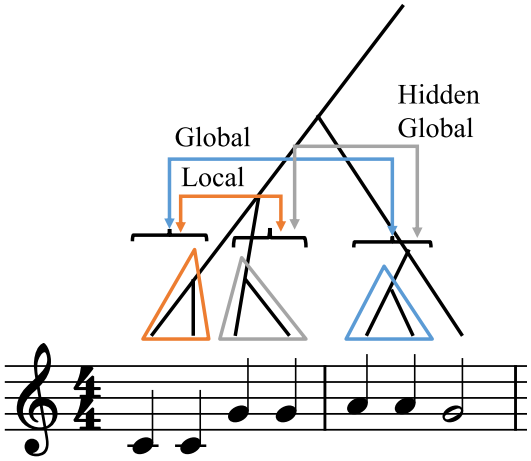

Fig. 1. Local and global relations in time-span tree

In this paper, we consider local and global analyzes of time-span trees. In detail, this paper reproduces the analysis and prediction of human music listening by taking missing time-span trees as inputs and complemented time-span trees as outputs. It enables local and global analysis and prediction using Sequence to Sequence (Seq2Seq) models and fill-in-the-blank tasks. The Seq2Seq model uses LSTM, and the recursive processing of LSTM enables analysis and prediction at the same time. Also, the input must be a tree structure. However, although previous research has discussed the analysis of tree structures[7, 8], using the tree structure itself as data has not been previously considered. In this research, based on the representation proposed in ON-LSTM, a tree structure input is realized by factoring in the hierarchical direction and horizontal directions. It predicts complemented time-span trees from missing time-span trees. In conclusion, this paper proposes local and global time-span tree filling tasks for analyzing time-span trees based on human music perception.

This paper is divided into five sections as follows. First, the definition of the time-span tree used is explained. Section 2 reviews previous studies using time-span tree and deep learning models for tree-structured data. Section 3 details the dataset and data augmentation and embedding used in this paper. Section 4 introduces the Seq2Seq model used for the fill-in-the-blank task in this paper. Section 5 presents results from

experiments to predict subtrees from subtrees. Section 6 presents a discussion of the results. Section 7 describes conclusions and future prospects.

2 Prerequisites and Related Work

Hamanaka et al. [5] analyzed time-span tree obtained by GTTM using deep learning. At low-level boundaries, the accuracy is 0.03 points higher than the conventional method, and the metrical structure has almost the same performance as the conventional method. However, this method is analyzed based on the theory of GTTM, and it shows that it cannot be analyzed without analyzer of GTTM. Hamanaka et al. [6] realized time-span analysis and melody morphing using deep learning. The research makes it possible to analyze time-span trees by learning the order of melody reduction. However, it does not take a tree structure as input, and also requires a melody input.

Ordered Neurons LSTM (ON-LSTM) [7] showed a function called Cumax. This function allows deep learning to learn the composition of the hierarchical structure. Then, the paper demonstrated that the function is effective in unsupervised learning. It shows that it is similar to analyzing a tree structure by a person who does not know the rules. However, it is not extended from tree structure to learning tree structure. Pyraformer [8] investigated whether hierarchical attention mechanisms are effective for long-term dependencies on time-series data. In addition, Pyraformer reduced the amount of calculation compared to the conventional method. However, it requires a large amount of data, well exceeding the number of pieces of music that a human being can listen to in a lifetime.

In conclusion, there is no model that predicts a tree structure from another tree structure, and either rules are used, or a large amount of data is required to acquire the hierarchical structure. Therefore, it is necessary to research a model that predicts and analyzes tree structure with a small amount of data without knowing the theory, similar to people listening to music.

3 Create the Experimental Dataset

The size of the tree structure varies greatly both vertically and horizontally. This tendency is particularly strong in the case of binary trees. There are two cases of extreme tree structures. One is when every possible reduction always occurs at each layer. In this case, since there are many tones, the sequence length is long, but the number of layers is relatively small. On the other hand, there are cases where the reduction is done only once at each Layer. In this case, even if the number of tones is small and the sequence length is short, the number of layers increases in proportion to the number of tones. As a result, when a tree structure is used as input, the size of the data differs greatly. Therefore, by using different learning methods for the vertical and horizontal directions, the effects of each are reduced and, with the implementation of deep learning, learns rules related to time-span trees. In the following, four procedures for solving the subtree filling problem of a time-span tree with Seq2Seq.

3.1 Data Collection

A dataset analyzing time-span trees of 300 songs has been published [9]. The time-span tree of the dataset is 8 bars long. Time-span trees are excluded that are ambiguous and subject to multiple analyzes due to preference rules. Furthermore, excluding the time-span tree whose sequence is long and sparse, the number of songs analyzed in this research is 279. The number of minimum layers is 5 and the maximum is 10. Also, the number of minimum notes is 10 and the maximum is 80.

3.2 Splitting the Dataset and Data Augmentation with Transposition

279 songs are divided into 8 to 2, and divided into 223 songs as learning data and 56 songs as test data. After that, the learning data is split 8 to 2 into training data and validation data, resulting in 178 songs in training data and 45 songs in validation data. Additionally, as a data augmentation, the training and validation data pitches are changed +2, +4, +5, +7, +9, +11. This multiplies the original data by 7.

3.3 Vectorization of Time-span Tree

The tree structure is constructed so that it is easy for humans to understand. However, it is a format that is difficult to handle for Artificial Intelligence such as deep learning. Therefore, blockview was proposed by ON-LSTM [7] as a vector representation of the tree structure. This research extended blokview, which normally analyzes in parse trees, to time-span trees. This research extends time-span trees to blockview. Also, smaller parts are zero-padded. In detail, vectorization is performed so that the maximum lengths in the vertical and horizontal directions can be obtained.

In the original time-span tree, the duration becomes the total value as it is simplified. But in this research, it is not the total value in order to avoid the prediction becoming deterministic due to the duration. In addition, when using the total value, it is possible that the accuracy of the prediction will be greatly affected. The reason is that all durations that appear must be labeled, and with the current number of data, the data will be sparse. For these two reason, duration is not a total value in this research.

Fig. 2 shows the original time-span tree and the time-span tree vectorized by the blockview. Note id 1 wins the most, so only notes with note id are vectorized in fourth layer. Originally, the simplification of note id 4 and 5 is ambiguous whether it is the second layer or the first layer. But for the sake of simplification, it is the first layer that can be considered. For note id 1, the duration is 0.75, 1.5, 3, but as We said earlier, we can't consider all combinations of numbers, so we consider it only as 0.75.

3.4 Vertical Embedding

To predict the tree structure, the tree structure needs to be embedded into the latent space. Time-span trees vectorized by blockview are split vertically. This split vector is called a timestep. Timesteps have more similarity in elements as they go up in the layer. To take advantage of this feature, we use skip-thought [10]. Skip-thought captures a latent space from some sentence that predicts the sentences before and after it. Fig. 3

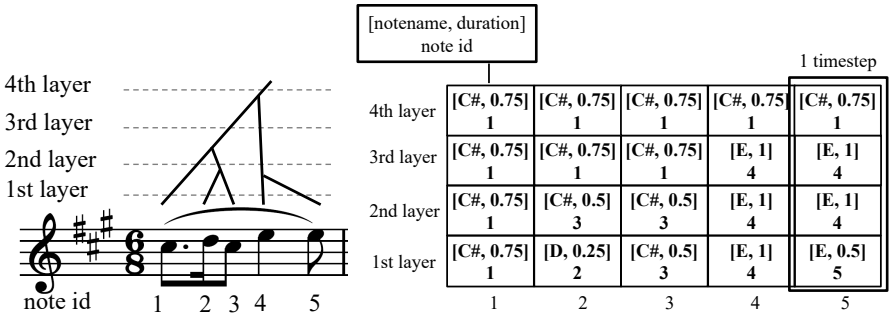

Fig. 2. The original time-span tree and corresponding blockview

shows the skip-thoght architecture for embedding timestep vectors. In addition, it is a Seq2Seq model that outputs a timestep vector with the latent space by skip-thogfht as input. Therefore, information above the maximum time-span must also be masked for a complete gap-filling task.

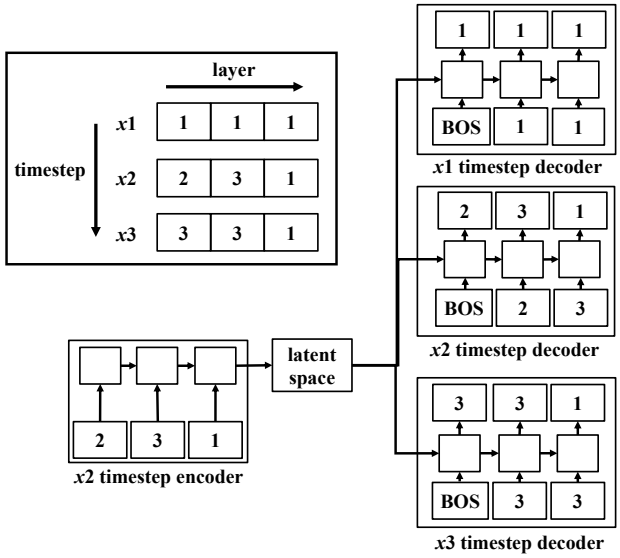

Fig. 3. Embedding x2 timestep by skip-thought

3.5 Data Augmentation and Create Fill-in-the-blank Tasks by Masking Blockview

By creating the experimental dataset, the training data is 1246 songs and the training data is 315 songs. Also, the test data is 56 songs. However, if one mask is applied to one song, the data is insufficient for deep learning.Therefore, we create all possible masked time-span trees. For example, if a time-span tree has n notes, n-1 time-span trees are obtained as masked time-span trees by subtracting the head of the time-span tree. Using Fig. 2 as an example, mask all but the maximum time-span with note id=1. Two examples are shown below.

If the note id is 2 in Fig. 4, the maximum time-span is only the first layer, so only the time step with note id 2 is masked. On the other hand, if the note id is 4 in Fig. 4, the maximum time-span is the second layer, so the timestep with note id 4 and the timestep with note id 5 that is reduced to 4 will be masked. In addition, in this research, since the latent space by skip-thought is input and the time step vector is output, it is necessary to mask the information above the maximum time-span tree in order to complete the blank filling task.

By masking, we obtained 43253 training data, 10780 validation data, and 1833 test data. For training and validation data, we obtained 34 times as many as original ones, and for test data, 32 times as many as original one.

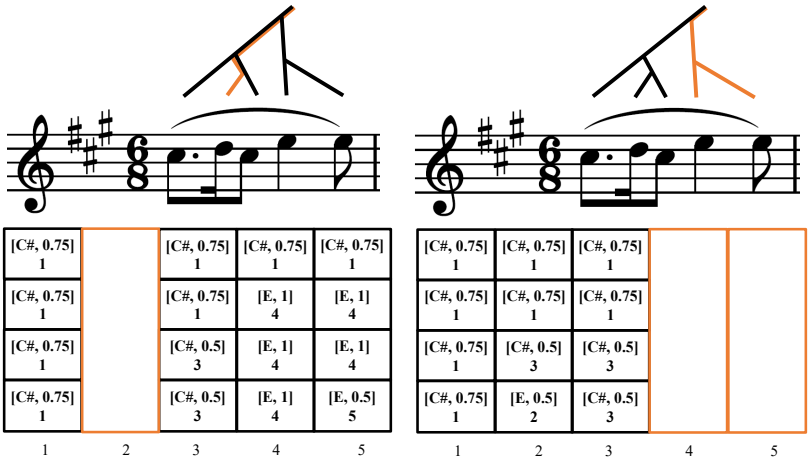

Fig. 4. (*left*) Masking the maximum time-span with note id 2, (*right*) Masking the maximum time-span with note id 4.

4 Overview of the Seq2Seq model

The purpose of this paper is to clarify the local and global relationships of time-span trees as a high-level analysis. To realize the implementation of this relationship, it is

necessary to do analysis and prediction at the same time. To analyze and predict at the same time, we propose a Seq2Seq model with LSTM. The Seq2Seq model realizes analysis and prediction at the same time by recursive processing of LSTM. In addition, it learns local and global relationships through a time-span tree fill-in task. Also, by using single-head attention, learn the relationship between local and global. Specifically, the decoder in the Seq2Seq model adds attention and computes the relationship between the masked time-span tree and the hole-filled time-span tree.

The Seq2Seq model used in this experiment takes skip-thought embedding as input and outputs a timestep vector as output. At this research, by solving the fill-in-the-blank task, it learns the global tree structure relationships. The LSTM used for the Seq2Seq model is bidirectional in the encoder and unidirectional in the decoder. Also, decoder use attention to help them learn global relationships. Fig. 5 shows the outline of the Seq2Seq model when attention is used.

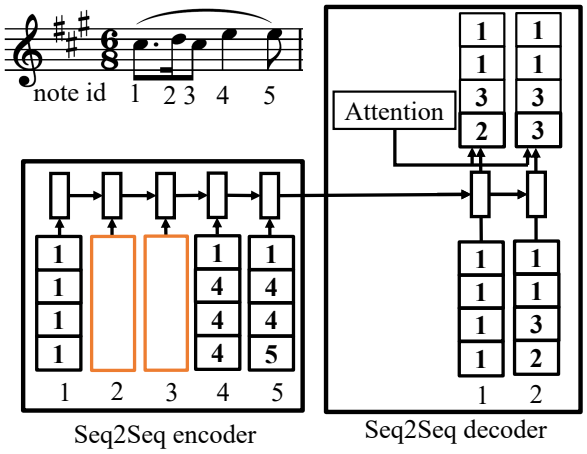

Fig. 5. Outline of Seq2Seq model using attention

Conventionally, attention takes all correspondences, but since fill-in-the-blank task, it does not calculate the masked timesteps. Fig. 6 shows the attention calculation in this research. This avoids the state where the correct answer data paired with the masked timesteps are visible due to attention. Self-correspondence is not used because the same problem can occur and affect the prediction of other masked time-span trees.

Also, this research, each note is a multi-hot vector by combining a one-hot vector for duration, a one-hot vector for octave, a one-hot vector for note name, a label indicating padding, and a label indicating mask. In addition, when outputting, padding labels are combined with duration, octave, and note name, and softmax is calculated as a one-hot vector. After that, the loss is calculated using the categorical cross-entropy as the loss function. Details of each category are shown in Table 1. The masked parts are note id 2 and 3 in Fig. 5, backpropagation is performed to reduce the loss of note id 2 and 3.

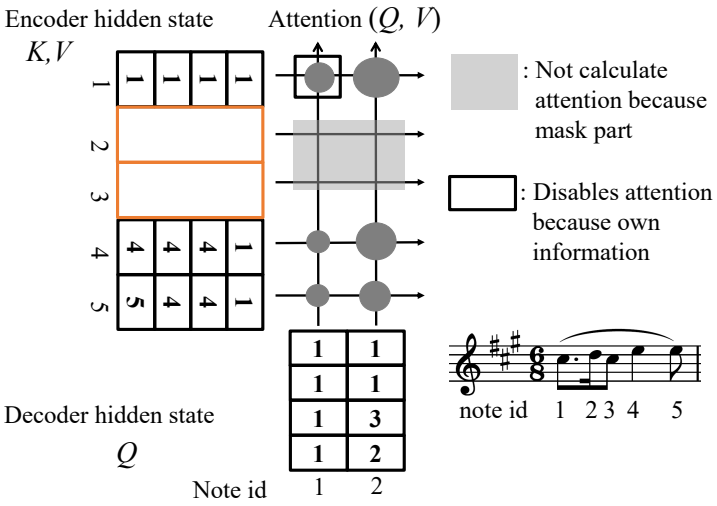

Fig. 6. Calculating the attention of the timestep vector in Fig. 5

Table 1. multi-hot vector ravel

category	contents of label	number
mask	mask or not	1
padding	BOS, EOS, padding for sequences, padding for layers	4
duration	0.125, 0.1667, 0.25, 0.3333, 0.375, 0.5, 0.625, 0.6667, 0.75, 0.875, 1.0, 1.167, 1.25, 1.333, 1.5, 1.625, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.5, 3.75, 4.0, 4.5, 5.0, 6.0	28
octave	4, 5, 6, 7, 8	5
note name	C, C♯, D, E♭, E, F, F♯, G, G♯, A,B♭, B	12

5 Experiment Results

Using the data set prepared in section 3, we show the results of training with the model in section 4. To see if attention is valid for local and global information, we compare it with a normal Seq2Seq model without attention.

5.1 Parameter tuning of Seq2Seq model

Deep learning requires learning with optimal parameters, but learning with all data requires a huge amount of computation for skip-thought and learning with Seq2Seq models. Therefore, we decided to find the optimal parameters for 178 training data and 45 validation data before data augmentation, and use those parameters when learning with all data.

To determine the parameters of the Seq2Seq model, the parameters were tested four dimensions: 200, 300, 400, 500, in the latent space of skip-thought and the hidden layer of the Seq2Seq model. Also, the seq2seq parameter tried three learning rates: 0.001, 0.0001, 0.00001. We trained Seq2Seq over 50 epochs and compared the loss function with the validation data. As a result, the loss value was lowest at 2.697 with a skip-thought dimension of 300, a hidden layer dimension of 200, a learning rate of 0.0001, and a batch size of 64. The top 10 losses are shown in Table 2. 6 of the top 10 losses no longer had a loss update within 10 epochs. Also, the loss was not updated when the learning rate was high, and the loss value was large when the learning rate was low.

Table 2. Validation loss value

thought dim	hidden layer	learning rate	batch size	epoch	validation loss
300	200	0.001	32	5	2.730
		0.0001	32	39	2.732
			64	29	2.697
400	400	0.0001	32	17	2.748
	500	0.001	32	3	2.765
500	200	0.0001	32	43	2.759
	300	0.0001	64	8	2.742
	500	0.001	32	2	2.760
			64	2	2.7576
		0.0001	64	7	2.761

As a result, for the fill-in-the-blank task, the best parameter skip-thought dimension was 300, the hidden layer was 200, and the learning rate was 0.0001. The batch size was 256, that it will be multiplied by 7.

5.2 Fill-in-the-blank task results

First, since the timestep was masked as a fill-in-the-blank task, we calculated the accuracy rate for the test data for each masking rate of the timestep including padding.

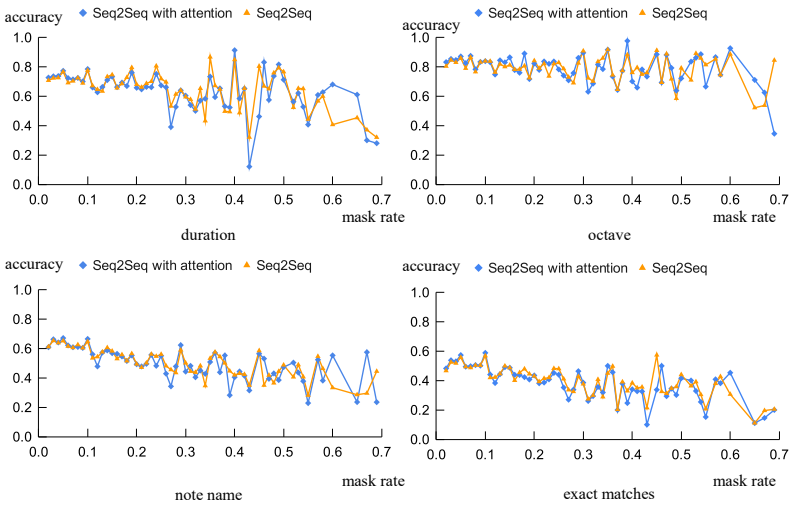

Fig. 7. Accuracy of mask rate in four conditions

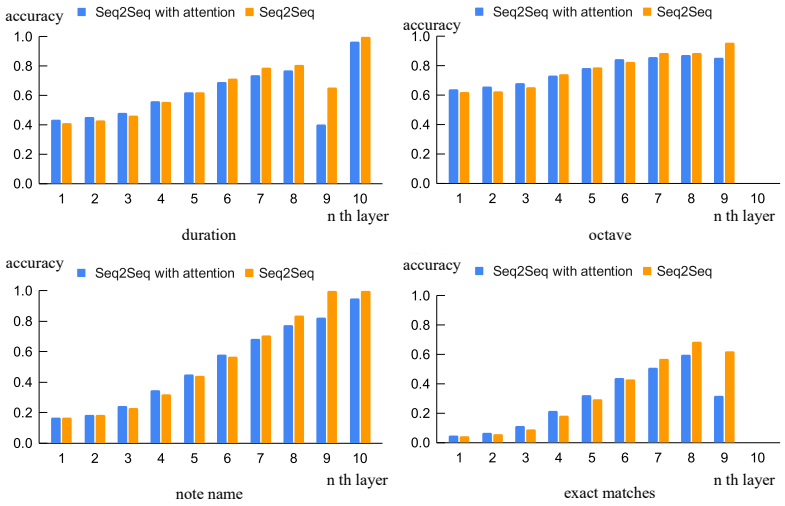

Fig. 8. Accuracy for each n th layer in four conditions

In other words, in a time-span tree with a maximum layer of 5, the remaining 5 layers are padding. The mask rate is calculated by dividing the masked timestep by the total timestep. Fig. 7 four accuracy for each mask rate. As a result, the prediction for the pitch name had the lowest accuracy without exact matchs. The duration was the worst when the mask rate was 0.43, the accuracy was 0.1208 with attention, and the accuracy was 0.3208 with the normal model.

Next, we show the result of whether the prediction was correct as a layer. Fig. 8 shows the four accuracy for each n th layer. In the graph, padding has been removed to show how well the predicted timesteps restore the original time-span tree. Therefore, the total number of predictions made decreases with each layer. As a result, we were able to confirm the pure accuracy for each layer. In all four results from first to fifth layer, Seq2Seq with atteniton is equal to or better than normal Seq2Seq. For sixth layer, Seq2Seq with attention gave better results for octaves and note names. However, normal Seq2Seq performed better in all results from seventh to tenth layer. For ninth layer, the duration was 0.4036 for Seq2Seq with attention and 0.6539 for normal Seq2Seq. The result was 0.3677 lower with attention and 0.1515 lower with normal, than eighth layer. In particular, Seq2Seq with attention, the accuracy was the lowest for all layers. Octaves were not predictable at all for tenth layer. Along with that, the exact matches was also completely unpredictable in the case of tenth layer.

6 Discussion

Let us discuss the following four issues.

- The parameter tuning of the Seq2Seq model : The top 10 losses were presented, and 6 of them completed training within 10 epochs. The learning has not converged from the general case of deep learning. It is necessary to devise more evaluation functions.
- The treatment of duration : In GTTM rules, simplification is performed using the total value of duraiton. Due to the number of labels, this study did not use total values. However, this method is not to learn the higher the hierarchy. To discuss this issue, we implement a loss function that recursively predicts the duraiton and compares the lengths.
- The data size is small : In fact, the most predicted factor was octave, and the next was duration. This is thought to be due to data augmentation for the pitch during learning. It is necessary to increase the total number of time-span trees and learn the data with a suitable distribution. In addition to the number of data, it is also necessary to discuss for labeling. This time, the note names are simply labels, but we should also consider the difference in how many notes are separated from each other.
- Using attention : The reason why there was no difference between Seq2Seq with attention and normal Seq2Seq is that attention was single-headed. Single-headed attention takes only one relationship. On the other hand, multi-head attention takes multiple relationships. Therefore, multi-head attention may improve subtree-to-subtree prediction.

7 Conclusion

We proposed learning a global tree structure by a fill-in-the-blank task of a time-span tree. As a result, Small subtrees within a time-span tree can now be predicted based on local and global relationships. However, the larger the subtree, the lower the accuracy. In addition, it is now possible to make predictions about the upper layers. However, the lower the layer, the more difficult the prediction, and the first layer could hardly be restored. Also, using attention did not show any significant improvement in accuracy.

The points to be improved in the future are as follows. Since the blockview is divided vertically as a timestep and the skip-thought is used as an embedded expression, it is possible that the constraints before and after are not completely predicted. This could be improved by changing the Seq2Seq decoder activation function to Cumax, which is used in ON-LSTM. Also, by taking the relationship between attentions like pyraformer, it may be possible to take the relationship in units of subtrees.

8 Acknowledgements

This work was supported by JSPS KAKENHI Grant number 21H03572.

References

1. Meyer, L.B.: Meaning in music and information theory. The Journal of Aesthetics and Art Criticism, 15(4), pp.412—424. (1957)
2. Egermann, H., Pearce, M. T., Wiggins, G. A., McAdams, S.: Probabilistic models of expectation violation predict psychophysiological emotional responses to live concert music. *Cognitive, Affective, & Behavioral Neuroscience*, 13(3), pp.533-–553. (2013)
3. Agres, K., Abdallah, S., Pearce, M.: Information-theoretic properties of auditory sequences dynamically influence expectation and memory. *Cognitive Science*, Vol. 42, pp.43—76. (2018)
4. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music, The MIT Press, Cambridge (1983)
5. Hamanaka, M., Hirata, K., Tojo, S.: deepGTTM III: Multi task Learning with Grouping and Metrical Structures. *the 13th International Symposium on Computer Music Multidisciplinary Research*, pp.161–172. (2018)
6. Hamanaka, M., Hirata, K., Tojo, S.: Time-span Tree Leveled by Duration of Time-span. *the 15th International Symposium on Computer Music Multidisciplinary Research*, pp.155–164. (2021)
7. Shen, Y., Tan, S., Sordoni, A., Courville, A.: Ordered Neurons: Integrating Tree Structures into Recurrent Neural Network. *In Proceedings of International Conference on Learning Representations*, New Orleans (2019)
8. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., Dustdar, S.: Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. *International Conference on Learning Representations*, Online, (2022)
9. GTTM Database, `https://gttm.jp/gttm/ja/database/`
10. Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-Thought Vectors. *Advances in Neural Information Processing Systems 28*, Montreal (2015)