A Melody Input Support Interface by Presenting Subsequent Candidates based on a Connection Cost

Tatsunori Hirai[⋆]

Komazawa University
thirai@komazawa-u.ac.jp

Abstract. In this paper, we present a melody input support interface that offers multiple pre-existing melody fragments as potential continuations for the melody being composed. The proposed interface utilizes the connection cost between melody fragments, based on the BiLSTM approach proposed by the author [1]. It provide subsequent candidate melodies or notes when the user encounters difficulties or needs fresh ideas during the melody composition process. Specifically, we consider a melody composition scenario in which the user inputs melodies onto a piano roll. We propose an interface that searches and presents subsequent candidate melodies or notes from a database comprised of existing melodies, based on the user's inputted melody. We conducted a user study on melody composition utilizing the proposed interface and assessed the effectiveness of the interface, as well as the quality of the generated melodies. The results confirmed the effectiveness of the proposed interface.

Keywords: Composition support; Connection Cost; LSTM

1 Introduction

Melody is a crucial element that characterizes a musical piece, and its creation is prioritized in music production. Melodies can exhibit a wide variety of characteristics, ranging from simple motifs repeated multiple times to intricate compositions that do not feature identical melodies from beginning to end. A common aspect among numerous musical pieces is the requirement for a melody to possess adequate length to constitute an entire song, compelling a composer to craft such a melody from the ground up. However, there is a constraint on the length of a melody that can be conceived at once, often resulting in the creation of only a small portion of the entire song at a time. A prevalent approach in melody composition, albeit with numerous exceptions, involves generating short, phrase-sized melodies and connecting them sequentially.

Generating short melodies through humming is relatively easy and is considered achievable even for individuals without expertise in music composition. Conversely,

[⋆] This work was supported by JSPS KAKENHI Grant Number JP19K20301 and JP23K17023.

crafting a melody for an entire song from start to finish is not something that everyone can do easily. Based on this observation, we consider that the difficulty in creating a melody primarily lies in effectively connecting short melody phrases. Consequently, we propose an interface designed to facilitate melody creation by presenting multiple candidate melodies that can follow the melody being created, utilizing the melody connection cost, a metric quantifying the naturalness of the connection between melody fragments.

By connecting short melody fragments conceived by an individual, it is possible to create longer melodies, potentially transforming a simple act of humming a tune into a more professional music production. Furthermore, if one can measure which melodies naturally connect together when creating mashup music comprising multiple tracks, irrespective of whether they are original or pre-existing, it could pave the way for supporting music production As an exploration of the potential for music production support, this paper examines an interface designed to support melody input using connection costs.

In recent years, deep generative models such as Music Transformer [2] and MusicVAE [3] have been proposed for melody generation, yielding high-quality results. Many of these approaches are categorized as “automatic composition” models, implying a significant machine contribution when users employ them for creative purposes. In this study, we investigate the potential of supporting melody composition while maintaining a balance between human creativity and machine involvement.

The melody connection cost employed in this interface is based on a previously proposed model by the author, which utilizes BiLSTM [1]. This model can also be adapted for automatic melody generation through minor modifications to the network configuration. However, in this study, we refrain from generating melodies and solely use the model to calculate the naturalness of connections between melodies. The objective is to develop a system capable of suggesting melodies that can be connected to the melody currently being produced, drawing upon a vast collection of existing melodies.

Our interface does not generate melodies; rather, it provides existing melodies when necessary. Consequently, our objective is to develop a support interface that functions similarly to predictive text input. Its use is not obligatory, but it can be employed when beneficial candidate options are presented. In the proposed interface, the subsequent candidate melodies are not machine-generated but are manually created melodies. Furthermore, the machine’s role is minimized, as the final selection of subsequent melody candidates is left to the user’s discretion.

2 Related Work

Bretan et al. proposed a melody generation technique employing existing melodies based on the connection cost of melodies, referred to as the unit selection method [4]. In this approach, new melodies are automatically generated by reusing pre-existing melodies. However, Bretan et al. did not focus on developing user-oriented support for music creation or associated interfaces. Furthermore, their method takes into account not only the connection cost between melodies but also their semantic relationships in order to narrow down the search space.

Cope also proposed an approach for generating new music by connecting existing melodies [5]. This method involves dividing a musical piece into small fragments, labeling each fragment according to its characteristics, and subsequently creating new music through the reuse and recombination of these fragments. The approach by Cope differs from the method presented in this paper as it relies on rule-based melody reconstruction rather than machine learning-based modeling.

The concept of employing existing melodies in music generation has been previously proposed. Pachet introduced a system called “The Continuator” that generates new melodies by dividing existing melodies into small fragments, modeling transitions between fragments using a tree-structured Markov chain, and searching for appropriate subsequent melodies from the training data [6]. Kitahara et al. proposed JamSketch, which generates improvised melodies in real-time using a genetic algorithm and existing melodies, based on the user’s rough outline of the melody input [7]. Although JamSketch does not utilize existing melodies in their original form, it is one example of utilizing existing melodies for melody generation.

The approach of generating new content by reusing existing content has been explored in various domains beyond music. For instance, it has been applied to image synthesis [8] and music video generation [9]. In this study, we focus on melody creation and propose an interface that utilizes the connection cost between melody fragments [1] to present existing melodies as candidates for subsequent melodies.

3 An Interface for Melody Input Support based on Connection Cost

Our interface is designed as a melody input support tool that utilizes the connection cost between melody fragments based on the BiLSTM proposed by the author [1]. A piano roll is commonly employed when composing a melody using a computer. Consequently, the input support interface in this study aims to facilitate melody creation utilizing a piano roll.

3.1 Basic Configuration of the Proposed Interface

The proposed interface is implemented as an additional feature on top of the conventional piano roll. Users can input notes by dragging the piano roll using a pen-style input tool. As fundamental functionalities, the interface incorporates quantization features for aligning the onset timing and length of notes, a function to move, modify, and delete input notes, and capabilities to play, pause, and stop the entered melody. The interface is designed for inputting melodies by note, employing a grid in the time direction using a 4/4 time signature, with four beats per bar and a 16th note as the smallest unit.

This interface offers all the fundamental features typically present in a standard piano roll, facilitating users to accomplish all the essential tasks for melody input. By incorporating a function that presents information based on connection cost, melody input can be supported.

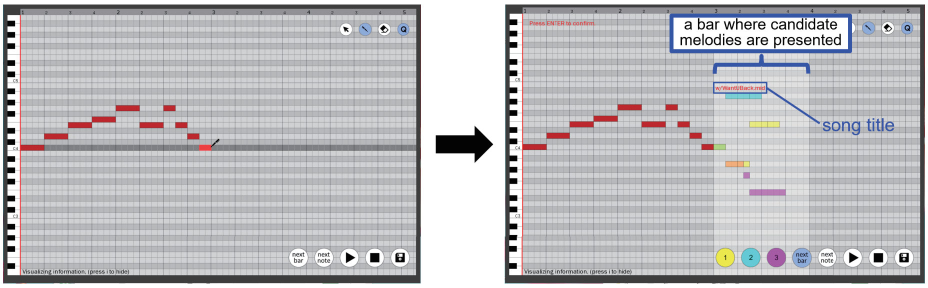
Visualizing information. (press i to hide)
next bar
next note
Press ENTER to confirm.
a bar where candidate melodies are presented
w/WantUBack.mid
song title
C5
C4
C3
1
2
3
next bar
next note
Visualizing information. (press i to hide)

Fig. 1. Melody candidate suggestion based on connection costs

3.2 Melody Candidate Suggestion based on Connection Costs

We propose a function that suggests subsequent melody candidates based on the connection cost between melody fragments. The proposed function searches a pre-existing melody database for the three fragments with the lowest connection cost that are most likely to continue the inputted melody, and presents them to the user as recommendations for melody continuation. Upon inputting one or more bars of melody using the piano roll and pressing the ”next bar” button, the interface calculates the connection costs between the last bar of the user’s inputted melody and pre-existing melodies in the database that are one bar in length. The interface then suggests three possible melody candidates for the user to continue their melody, based on this calculation. Fig.1 shows an example of the subsequent melody candidate presentation. The left side of the Fig.1 shows the melody that the user manually inputted, while the right side of the Fig.1 shows the result screen after pressing the “next bar” button. The bright background in the piano roll indicates the bar where the subsequent melody candidates are presented. Users can listen to the three suggested subsequent melody candidates and select one to connect with their input melody. If users find a suggested melody that they like, they can incorporate it into their composition.

The name of the MIDI file from which the suggested melody candidates were extracted is displayed on the piano roll. If a candidate is selected, the corresponding song title will continue to be displayed on the melody of the corresponding bar. After selecting a melody from the presented candidates, the user can edit it further as with a typical piano roll interface. If the user doesn’t like a certain part of the candidate melody, they can modify it to fit their own image while keeping the original style. At this time, the user is not required to adopt the presented melody into their own composition, so it can be used only as a reference when the user gets stuck in their composition process. This feature is positioned to assist users only when necessary, as it is not a mandatory function.

Although automatic composition methods that generate subsequent melodies have been previously proposed, a notable aspect of our interface is that the suggested melodies are based on existing melodies, which are manually created rather than generated automatically. With this function, the user can add preferred melody data to the database and search for melodies that are more likely to be connected to the current melody from a large number of existing melody dataset. They can then adopt these

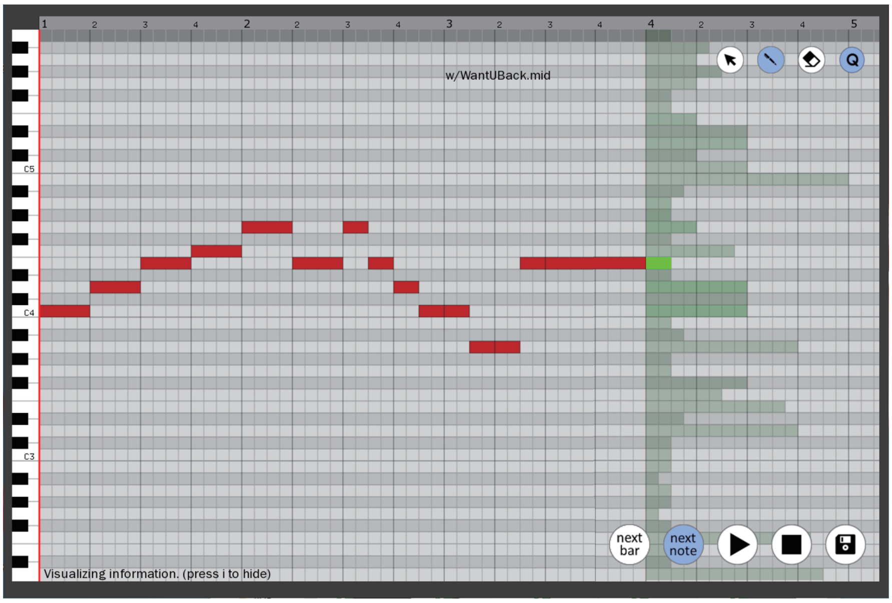

Fig. 2. Visualization of subsequent note candidates

melodies as part of their own composition. The title of the original melody is displayed at the top of the corresponding melody, making it possible to create the user's own melody while inheriting and citing existing melodies.

When the "next bar" button is pressed, inference is performed in the background to calculate the connection cost between the input melody and the melodies in the database. Therefore, the more melodies there are to search, the longer the wait time until candidate melodies are presented. Currently, when searching for candidates for 10,000 bars, it takes approximately 20 seconds on a machine with 32.0GB memory, Intel Core i9-1088H 2.40GHz, and NVIDIA GeForce RTX 2060. When using this feature, shorter wait times are desirable as they allow for more attempts to be made, and faster feedback can be obtained. The waiting time can be shortened by improving the implementation, and reducing it further is our future challenge.

3.3 Visualization of Subsequent Note Candidates

When calculating the melody connection cost, the validity of note-level connections is also considered, and by visualizing it during melody input, a user can examine what would be appropriate as the next input note. Pressing the "next note" button reveals the candidates for subsequent notes, including the type and likelihood of notes that are likely to follow the last note the user inputted. Fig.2 shows how the subsequent note candidates are visualized.

As shown in Fig.2, the interface visualizes which pitch and duration the user would be preferable to input as the next note after the last note they inputted. This visualization is based on the frequency of note transitions in the melodies of the dataset used to train the original melody connection cost calculation model. Therefore, it simply shows more common note transitions in a darker green color. Since many existing melodies have frequent transitions to the same pitch, this function often suggests notes of the same pitch as the most probable candidates. It should be noted that this is simply an information visualization, and users are not obligated to input the next note based on this information. This information can serve as a reference when transitioning to less common notes, and is intended as a suggestion to the user while they actively input the melody.

The proposed interface provides two functions to assist with melody input: suggesting subsequent melody candidates by bar and by note. The suggestion of candidates is entirely optional, and both functions are designed to be utilized only when the user needs them.

4 User Study

A user study is conducted to evaluate the effectiveness of the proposed melody input support interface.

4.1 Conditions of the User Study

We conducted a user study with four participants who used the proposed melody input support interface. Each participant completed six melody input trials, three with and three without using the function for suggesting candidate melodies. After each trial, participants responded to a questionnaire to evaluate the system's effectiveness. Before starting the user study, the author demonstrated how to operate the interface to the participants, and they were given the opportunity to try it out after learning the basic operation method. We also provided a document that explained the details of each button, which participants could refer to if they were unsure of how to operate the interface during the trials. The participants' musical experience for this user study was as follows:

- User A: Less than 1 year of music experience, no experience in DTM (desktop music: music production software), and some experience in composing songs at a level of humming.
- User B: No music experience, no DTM experience, no composition experience.
- User C: Over 10 years of musical experience, experience with DTM, and some experience in composing songs at a level of humming.
- User D: No music experience, no DTM experience, no composition experience.

Participants were asked to input short melodies consisting of 2 to 4 bars with the proposed interface, and the interface was evaluated through multiple trials. During the trials where the melody candidate suggestion function was utilized, participants were instructed to use the function within a 4 bars, while the subsequent note candidate suggestion function was optional and used only when necessary. For each participant's six trials, the subsequent melody candidate suggestion function was used on even-numbered trials, alternating between trials with and without its use. Trials excluding the melody candidate suggestion were utilized as our baseline. In the baseline trial, participants inputted a melody of 2 to 4 bars into the piano roll interface without any guidance.

The database used for the melody candidate suggestion function consisted of 10,000 bars of melody randomly extracted from test data that were not used for training the connection cost calculation model. When the suggestion function is used under these conditions, it takes about 20 seconds to process.

In addition to assessing the interface, upon completion of all trials by the participants, we further evaluated the melodies themselves. Each participant's set of 6 melodies was reviewed by three other participants, who were not the original creators, for evaluation.

4.2 Evaluation Items

Participants were asked to evaluate each melody creation trial based on the following four evaluation criteria.

1. Able to create a desired melody
2. Able to create a unexpected melody
3. Able to create a satisfactory melody
4. Able to create melodies easily

The melody was created six times in total, with three times using the melody candidate suggestion function and three times without using it. After completing each melody, the participants were asked to rate the four evaluation criteria mentioned above on a 4-point scale, with options "1: Does not apply", "2: Somewhat does not apply", "3: Somewhat apply", and "4: Apply".

After the 6 trials and responses to the evaluation items were completed, an overall evaluation was conducted. For each subsequent melody and subsequent note candidate suggestion function, participants were asked to rate their effectiveness on a 4-point scale: "1: Not effective", "2: Somewhat not effective", "3: Somewhat effective", "4: Effective". Furthermore, regarding the subsequent note candidate function, each participant was asked to evaluate the degree of use of the optional subsequent note candidate function, which was evaluated in four levels: "1: Almost never used", "2: Rarely used", "3: Used several times", "4: Used frequently". Finally, participants were asked to give their general opinions and feedback in an open-ended format.

All user trials were recorded with screen captures, and the duration of each trial was measured. Furthermore, the influence of the feature on the time needed to create a melody was assessed.

The evaluation of all melodies created by the participants in the user study was conducted by asking them to rate each melody on a 4-point scale, ranging from "1: not a good melody", "2: not a very good melody", "3: somewhat a good melody", to "4: a good melody". Additionally, the evaluation was conducted by the remaining three participants of the user study who listened to each melody without knowledge of how it was created.

Table 1. Evaluation results of melody creation trials

	Evaluation items			
	(1)	(2)	(3)	(4)
without candidate suggestion	2.17	2.25	2.08	2.42
with candidate suggestion	**3.25**	**3.58**	**3.17**	**3.75**

Table 2. Evaluation results of each function's effectiveness

	average evaluation score
Effectiveness of melody candidate suggestion	3.75
Effectiveness of note candidate suggestion	2.67
Frequency of using note candidate suggestion	1.5

All evaluation items were rated on a 4-point scale, where higher ratings denote better performance. The intermediate value is 2.5, with ratings above this value indicating a positive outcome.

4.3 Evaluation Results

The results of the user study are presented in Table 1, 2, and 3. Table 1 shows the evaluation results for each melody creation trial. It presents the average evaluation scores separately calculated for the presence and absence of the candidate suggestion function. Table 2 presents the evaluation results regarding the effectiveness of the candidate suggestion function after all trials were completed. It shows the average evaluation values for each item. Table 3 shows the evaluation results of the six melodies created by each participant, as evaluated by the remaining three participants. It shows the average evaluation values for all six melodies produced by each participant, including the average score with/without the candidate suggestion function. The evaluation scores range from 1 to 4, with higher values indicating better performance.

Based on the results presented in Table 1, all evaluation items received higher scores when using the subsequent melody candidate suggestion function compared to when it was not used. Notably, the use of the candidate melody suggestion function resulted in higher scores even for the evaluation item "able to create a desired melody." These results imply that the presented candidate melodies are more aligned with the melody that users imagine. Specifically, for participants who were creating a melody with piano roll for the first time, it appeared challenging to compose musically pleasing melodies. In such a situation, the melodies suggested by the candidate suggestion function are actual melodies that possess musical sense. Therefore, it is inferred that the support provided by the function fulfilled the users' requirements and facilitated them in achieving their melody creation goals.

As shown in Table 2, the average evaluation score for the subsequent melody candidate suggestion function's effectiveness was 3.75, with all four participants indicating that it was effective. In contrast, the note candidate suggestion function's average evaluation score for effectiveness was 2.67 and was not evaluated as particularly effective. In terms of usage frequency, three out of the four participants reported that they "almost

Table 3. Evaluation results of composed melodies

User	Evaluation score					
	without candidate suggestion			with candidate suggestion		
A	3.33	2.33	2.00	3.67	3.33	2.67
B	1.67	2.33	2.33	2.33	2.33	3.33
C	3.00	3.33	3.00	3.00	3.67	3.33
D	1.67	1.67	3.33	3.00	2.67	3.00
average	2.50			**3.03**		

never used" the feature, indicating that it did not contribute significantly to melody creation support.

Based on the evaluation results of the melodies produced by the four participants, as presented in Table 3, the melodies created using the melody candidate suggestion function received higher overall ratings than those created without using the function. The quality of the created melodies varied among users. For instance, user C, who had the most musical experience, received evaluation scores of 3 or higher for all of their created melodies. Examining the evaluation values for each melodies based on whether they used the function or not, it can be seen that every user was able to create higher-quality melodies by using the function. These results indicate that the interface support has improved the quality of the melodies produced.

The following are some of the comments obtained through the open-ended section at the end of the trial[1].

- I would like the system to propose other melodies when I don't like the suggested melody.
- The note suggestion function kept suggesting the same notes.
- I was glad that the created song didn't become monotonous because the system suggested melodies that I wouldn't have thought of myself.
- After repeating the process, I gained a sense of what makes a melody work and felt that as I became better at creating melodies, the suggested melodies also improved.

The feedback obtained suggests that the interface provided a certain level of useful assistance; however, there is still room for improvement in the subsequent note suggestion function. We intend to incorporate the feedback received to enhance the interface in the future.

Finally, we evaluated the impact of using the melody suggestion functions on the time required for creating melodies. Table 4 shows the time required for all six melody creation trials for each user. In the condition with the melody suggestion function, the waiting time for suggestions (approximately 20 seconds per use) was also included in the total time. Users with less experience tended to use the candidate suggestion function multiple times, resulting in longer overall required times due to the waiting times that occurred each time. Moreover, the time required for comparing and listening to the three proposed melody candidates also added up to the required time. Therefore, it can be concluded that the current interface does not contribute to the efficiency of

[1] The comments originally provided in Japanese have been translated into English by the author.

Table 4. Evaluation of the duration required for each trial

	Time required for trial					
User	without candidate suggestion [s]			with candidate suggestion[s]		
A	681	184	233	450	663	551
B	138	134	244	208	261	231
C	166	124	193	181	222	216
D	218	191	181	297	238	368
average	**223.9**			322.2		

melody creation in terms of time. We aim to address this issue by improving the system speed and providing more suitable candidate melodies based on user needs in the future.

Through this user study, it became apparent that users improved their melody creation skills as they repeated the trials. Additionally, some users gained an understanding of what kind of melodies to input to receive better candidate suggestions. These findings suggest that, like traditional music production tools, repeated use of this tool can lead to greater proficiency, making it more convenient to use. The observation that humans adapt their behavior to the tool suggests the potential for collaboration between artificial intelligence technology and human music creation, making it an intriguing research topic for future studies.

5 Discussion

In this chapter, we discuss the potential and concerns of the interface introduced in Section3.2, which enables the reuse of existing melodies.

As mentioned in Chapter1, this interface was developed with the idea that if the act of inputting short melody phrases such as humming can be connected to the creation of longer melodies for an entire song, anyone can easily engage in music production. The interface is designed to support such endeavors, and the results of the user study in Chapter4 demonstrate the effectiveness in melody creation.

When reusing existing melodies, it can encourage the reuse of other people's creative works, which can be both positive and negative. Creative activities are often inspired by the works of others, and in music, for example, it is a legitimate practice to compose based on chord progressions of songs created by others. While it is difficult to deal with melodies and not permitted to use them as is, paying homage to past music by incorporating someone else's melodies into one's own work is a common practice. Short units such as a single bar have countless examples of songs that share melodies with other works. Sampling has emerged as a well-established musical genre and technique that involves incorporating segments of pre-existing music or sounds into one's own compositions. Our interface can be viewed as an interface that enables the direct sampling of melodies.

When using the function in our interface to suggest subsequent melody candidates based on existing melodies, the original song file name is displayed on the piano roll, providing an opportunity to credit the reused music information in the final composition. This allows for the creation of works that include citations, akin to the culture of fan fiction. However, the interface not only enables reusing melodies as they are but also re-editing them to fit one's own melody, posing a challenging issue from a copyright perspective on how to treat a reused melody that no longer retains its original form.

Using the proposed interface, one can extract melodies from short phrases previously created by oneself, in a manner akin to predictive text input, even without utilizing others' works as the database. By accumulating many short phrases on their own, users can conveniently extract their own melodies. As this process involves reusing materials created by oneself, there are no rights-related issues. We anticipate that the proposed interface will continue to serve as a useful tool when employed in this manner.

The proposed interface opens up new possibilities for collaborative music creation among multiple creators. Drawing inspiration from the way short sentences are retweeted and attached to other tweets on Twitter, we envision the possibility of expanding the system further by incorporating a mechanism that facilitates the reuse of short melody phrases shared by multiple users on social networking services (SNS). Such an approach would enable the construction of a single composition through the amalgamation of diverse phrases contributed by numerous users. This could lead to a future where someone's casual humming could be incorporated into a professional musician's new song.

Sound libraries such as Splice[2] offer numerous publicly available short audio materials that are utilized by creators worldwide as components of their works. Just as there are cases where lyrics are completed by collecting words submitted by fans and having professional artists write the final version, a collaborative production approach can also be applied to musical elements such as melodies. The proposed interface is one example of how such a production style can be implemented.

6 Conclusion

In this paper, we proposed an interface that supports melody input by presenting candidate melodies based on the connection cost between melody fragments. We conducted a user study to evaluate the effectiveness of the proposed interface for assisting melody input and confirmed its effectiveness by evaluating melodies created by users using the interface.

The proposed interface enables users to combine short melody fragments to construct longer melodies, seamlessly incorporating melodies created by themselves or other users as necessary. It includes a function similar to culture of fan-created content, allowing users to credit the sources of melodies used. This is particularly important since there is no clear legal definition of the maximum length of a melody that can be reused without infringing on copyright law. However, additional deliberation is required to judge whether edited melodies are also permissible for use. This interface can be used without infringing on any rights issues if users utilize melodies that they have previously created. In such cases, there are no copyright infringement issues as it involves reusing one's own material.

[2] https://sounds.splice.com/

A potential future direction for this research is to improve the response speed of the interface. To present candidate melodies for the subsequent phrase, the interface needs to perform inference to calculate connection costs between the input melody and all melodies in the dataset. Consequently, the current waiting time to compute the connection cost between the input melody and the 10,000-bar search candidates is approximately 20 seconds. Bretan's unit-based melody generation [4] narrows down the search space by utilizing the semantic relationship between melodies. Preprocessing, such as this, is crucial for enhancing processing speed. In the future, we aim to enhance the functionality of this interface to make it more practical and develop it into a tool that can be used with actual DAW software in formats such as VST plugins.

References

1. Hirai, T., Sawada, S.: A Method for Calculating Melody Concatenation Cost based on BiLSTM. Journal of Global Media Studies, Volume 31, pp.55–64 (2022)
2. Huang, A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A.M., Hoffman, M.D., Eck, D.: Music Transformer: Generating Music with Long-Term Structure. In: Proceedings of the International Conference on Learning Representations (2018)
3. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A hierarchical latent vector model for learning long-term structure in music. In: Proceedings of the 35th International Conference on Machine Learning, pp.4364–4373 (2018)
4. Bretan, M., Weinberg, G., Heck, L.: A unit selection methodology for music generation using deep neural networks. In: Proceedings of the International Conference on Computational Creativity (2017)
5. Cope, D.: One approach to musical intelligence. IEEE Intelligent Systems and their Applications, Volume 14, No.3, pp.21–25 (1999)
6. Pachet, F.: The continuator: Musical interaction with style. Journal of New Music Research, Volume 32, No.3, pp.333–341 (2003)
7. Kitahara, T., Giraldo, S., Ramírez, R.: JamSketch: Improvisation Support System with GA-Based Melody Creation from User ’ s Drawing. In: Proceedings of the International Symposium on Computer Music Multidisciplinary Research, pp.509–pp.521 (2017)
8. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics, Volume 28, No.3 (2009)
9. Nakano, T., Murofushi, S., Goto, M., Morishima, S.: DanceReProducer: An Automatic Mashup Music Video Generation System by Reusing Dance Video Clips on the Web. In: Proceedings of the 8th Sound and Music Computing Conference, pp.183–pp.189 (2011)