ACCOMONTAGE2: A COMPLETE HARMONIZATION AND ACCOMPANIMENT ARRANGEMENT SYSTEM

Li Yi[1,2] **Haochen Hu**[1,2] **Jingwei Zhao**[3] **Gus Xia**[1,2]
[1] Music X Lab, NYU Shanghai [2] MBZUAI
[3] Institute of Data Science, NUS
ly1387@nyu.edu, hh1933@nyu.edu, jzhao@u.nus.edu, gxia@nyu.edu

ABSTRACT

We propose AccoMontage2, a system capable of doing full-length song harmonization and accompaniment arrangement based on a lead melody.[1] Following AccoMontage, this study focuses on generating piano arrangements for popular/folk songs and it carries on the generalized template-based retrieval method. The novelties of this study are twofold. First, we invent a harmonization module (which AccoMontage does not have). This module generates structured and coherent full-length chord progression by optimizing and balancing three loss terms: a micro-level loss for note-wise dissonance, a meso-level loss for phrase-template matching, and a macro-level loss for full piece coherency. Second, we develop a graphical user interface which allows users to select different styles of chord progression and piano texture. Currently, chord progression styles include Pop, R&B, and Dark, while piano texture styles include several levels of voicing density and rhythmic complexity. Experimental results show that both our harmonization and arrangement results significantly outperform the baselines. Lastly, we release AccoMontage2 as an online application as well as the organized chord progression templates as a public dataset.

1. INTRODUCTION

Accompaniment arrangement is a difficult music generation task involving structured constraints of melody, harmony, and accompaniment texture. A high-quality arrangement could help with various downstream tasks and applications, such as compositional style transfer [1], automatic accompaniment [2], and score-informed source separation and music synthesis [3].

As one of the most promising arrangement systems, AccoMontage [4] uses a generalized template-based approach to first search for roughly-matched accompaniment phrases as the reference and then re-harmonize the selected reference via style transfer. It generates much more coherent results than purely learning-based algorithms, especially for full-length song arrangements.

However, AccoMontage is not yet a "complete" accompaniment generation system in the strict sense, as it still calls for chord input from users and cannot harmonize a melody. To this end, we develop AccoMontage2, a system capable of full-length song harmonization and accompaniment arrangement based on a lead melody by equipping AccoMontage with two extra components: 1) a novel harmonization module, and 2) a graphical user interface.

The main novelty of our system lies in the harmonization module. We first collect a high-quality chord progression dataset and re-organize the phrases with respect to different styles to serve as reference templates. Then, we use dynamic programming (DP) to generate structured and coherent chord progressions given a query lead melody with phrase annotation. Specifically, the DP algorithm optimizes a multi-level loss function consisting of three terms: 1) a micro-level loss for note-wise melody-chord matching, 2) a meso-level loss for phrase-template matching, and 3) a macro-level loss for the whole-piece coherency. The first term evaluates the dissonance between the melody and the candidate chords. The second term prefers chord progressions with the same length as the target melody phrases. The third term computes how well the candidate phrases connect with each other to form an organic whole. Experimental results show that both our harmonization and arrangement results significantly outperform the baselines.

In addition, we develop a graphical user interface which allows the user to select different styles of chord progression and piano texture. Currently, chord progression styles include R&B, Dark, Pop-standard, and Pop-complex. Piano texture styles include several levels of voicing density and rhythmic complexity.

We release the AccoMontage2 as an online application[2] as well as the organized chord progression templates as an open-source dataset.

In brief, the contributions of our paper are as follows:

- A complete system for full-length song harmonization and accompaniment arrangement;
- An effective harmonization algorithm with state-of-the-art performance;

[1] Codes and dataset at https://github.com/billyblu2000/accomontage2.

[2] Online GUI link at https://billyyi.top/accomontage2.

Attribution: L. Yi, H. Hu, J. Zhao, and G. Xia, "AccoMontage2: A Complete Harmonization and Accompaniment Arrangement System", in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, Bengaluru, India, 2022.

- A graphical user interface for controllable piano accompaniment generation.

2. RELATED WORKS

2.1 Melody Harmonization

Melody harmonization refers to the task of generating a harmonic chordal accompaniment for a given melody [5, 6]. It has been typically formulated as a prediction task, *i.e.*, to predict a sequence of chord labels conditioned on the lead melody. Recent mainstream methods range from hidden Markov models [7, 8] to deep neural networks [9–11]. Such models are typically trained to fit a groundtruth melody-chord mapping, but do not account for the fact that one melody can be harmonized with various styles in terms of genre, chord complexity, *etc*. In fact, the current state-of-the-art models [10, 11] only support simple triads and up to a few common seventh chords. Also, predictions are made locally, where neither phrase-level progression nor inter-phrase structures are explicitly considered.

In this paper, we re-formulate melody harmonization with a novel template matching approach. The usage of existing templates for music generation has been a popular idea. Existing template-based methodologies include learning based unit selection [2,12], rule-based score matching [13,14], and genetic algorithms [15]. In our case, we match the lead melody with chord templates from a library based on rule-based criterion and subject to user control. Such an idea is inspired by the fact that music producers tend to pick up off-the-shelf chord templates instead of harmonizing from scratch. In addition, they also have control on what style of the chords to use.

Existing template-matching attempts for harmonization typically focus on half-bar level [16]. In contrast, our model deals with phrase-level matching. We design three loss terms that measure melody-chord fitness at note-wise, intra-phrase, and inter-phrase levels, respectively. Our chord library is finely organized, supporting up to ninth chords with voice leading and various genres. Our model can therefore generate structured and coherent full-length chord progressions with different styles.

2.2 Accompaniment Arrangement

The task of accompaniment arrangement aims to generate an accompaniment conditioned on a given lead sheet (*i.e.*, a lead melody with chord progression). The quality of arrangement is related to chordal harmony, texture richness, and long-term structure. For this task, existing learning-based models often do well in harmony and texture, but are less capable of long-term generation [1, 17–21]. Previous template-matching models can easily maintain long-term structures, but suffer from fixed elementary textures and often fail to generalize [13–15].

To break such a dilemma, the AccoMontage system [4] introduces a generalized template-matching methodology, where phrase-level accompaniment templates are first searched by rule-based criterion, and then re-harmonized via deep learning-based style transfer. The search stage and the style transfer stage each optimize high-level structure and local coherency, thus guaranteeing the arrangement of high-quality accompaniment.

In this paper, we integrate our harmonization module with AccoMontage and upgrade it to a complete accompaniment generation model. An input melody is first harmonized with stylistic chord progression and then arranged with piano textures. With an additional GUI design, our model offers a flexible degree of control, including harmony styles and texture complexity.

3. METHODOLOGY

The system diagram of our AccoMontage2 system is shown in Figure 1. It can achieve full-length song harmonization and accompaniment arrangement. The input of the system is a query lead melody with phrase annotation. The harmonization module will first harmonize the melody. The generated chord progression will be sent into AccoMontage together with the original melody to arrange accompaniment. Lastly, a GUI is provided for users to adjust chord and accompaniment styles.

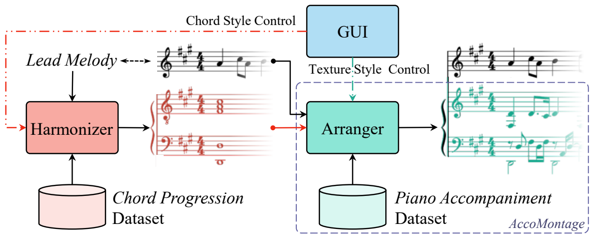


Figure 1: Diagram of AccoMontage2 system.

For the rest of this section, we first introduce the structure of the dataset and how we re-organize it in Section 3.1. Then we describe the harmonization module in Section 3.2. After that, we provide an overview of the AccoMontage system in Section 3.3. Finally, we show how the GUI is constructed to enable style controllability in Section 3.4.

3.1 Dataset Curation

A self-collected chord progression dataset is used as the reference templates for our harmonization algorithm. We create the dataset based on an existing chord progression collection [22] that contains 64,524 MIDI files, most of which are chord progression tracks with different style labels. The original dataset [22] cannot be adapted to our model directly for several reasons. First, some tracks are not pure chord progression, containing mixed melody segments. Second, many style labels are unnecessary. For example, two different styles may have similar musical elements that are hard to differentiate. Third, there are redundant progressions only different in their keys.

To solve the problems above, we process and re-organize the dataset as follows. First, we remove the MIDI files that contain melody segments and complex rhythmic textures. Second, based on our subjective assessment, we manually define a style mapping function to map the original style labels to newly defined ones. [3] The newly defined styles are: “Pop-standard”, “Pop-complex”, “Dark”, and “R&B”. While Pop-standard contains only triad chords, Pop-complex contains seventh, ninth, and sometimes even more chromatic ones. Note that some templates do not have an original style label and are thus labeled as “Unknown”. Third, we remove all the redundant progressions.

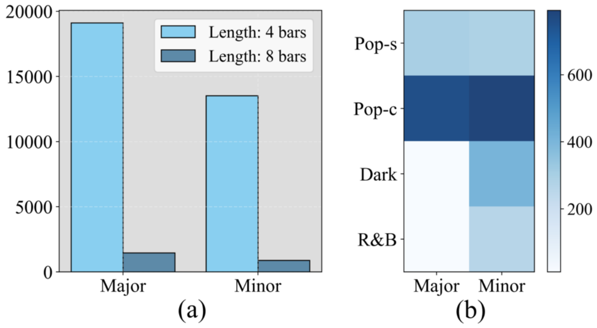


Figure 2: Statistics of the curated dataset.

The final curated dataset contains 5762 pieces of chord progression templates. Each template has 3 additional labels: 1) The length label. Templates are either of 4 bars or 8 bars in length in our dataset. Figure 2 (a) shows a distribution of the length of the templates. 2) The mode label. We currently only label the mode of the templates as either major or minor. 3) The style label. As mentioned above, four styles in total are acquired. The distribution of different styles among modes are shown in Figure 2 (b).

Finally, we represent the reference template space as a collection of tuples:

$$r = \{(r_m^{\text{chord}}, r_m^{\text{label}})\}_{m=1}^{N}, \tag{1}$$

where r_m^{chord} and r_m^{label} are the chord progression and the labels of the i^{th} reference template; N is the volume of the reference space. r_m^{chord} can be seen as a sequence of chords. We quantize and sample the chords at 8th notes from the original chord progression track. Each chord is represented as its root note and a label to indicate whether the corresponding triad is a major triad or a minor triad.

3.2 Harmonization Module

We design a harmonization model that generates structured and coherent full-length chord progression for a given lead melody with phrase annotation. The model takes a multi-phrase melody as input and outputs a list of optimal chord progression identities. Each identity contains a group of progressions that have the same Roman numeral sequence (*e.g.*, I-vi-ii-V) but different styles (*e.g.*, Pop-standard or R&B) and are up for the user to choose. The model considers three levels of losses that involve melody-chord correspondence and is optimized by a dynamic programming (DP) algorithm. Specifically, the DP algorithm optimizes a multi-level loss function consisting of three terms: 1) a micro-level loss for note-wise melody-chord matching, 2) a meso-level loss for phrase-template matching, and 3) a macro-level loss for the full piece coherency.

3.2.1 Micro-level loss

The micro-level loss L_{mic} computes the level of dissonance between a melody phrase and candidate progressions note by note. We mainly consider the interval between a melody note and the root of the chord in the corresponding position. The more dissonant the interval is (tritone, minor seconds, etc.), the higher the loss. On the other hand, the more harmonious the interval is (unison, perfect fourth, perfect fifth, etc.), the lower the loss. In addition, the mode of the piece and the harmonic function of the chords will also affect the dissonance level. We refer to the “rank order of consonances and their degree of recurrence” [23] and design the micro-level loss shown in Figure 3, where matrix (a) is for major mode and matrix (b) is for minor. For both matrices, each row represents the degree of the chord (we only consider diatonic chords, *e.g.*, 1 for tonic and 5 for dominant) and each column represents the interval between the melody note and the key center. In Figure 3, the darker the shade, the more dissonant we consider the pair of the melody note and chord is, in which case we set a higher micro-level loss.

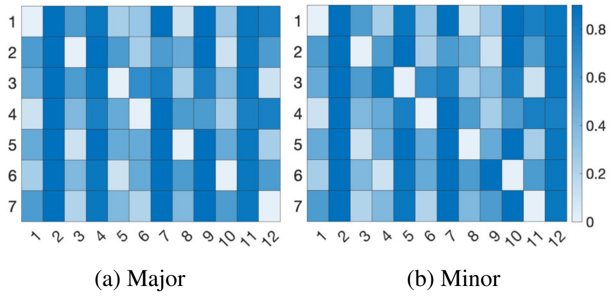


Figure 3: Micro-level loss indicating note-wise melody-chord dissonance.

For the i^{th} phrase, L_{mic} is computed by summing up the micro-level loss note by note and then dividing it by the length of the phrase. We further normalize L_{mic} to the range of $[0, 1]$.

3.2.2 Meso-level loss

The meso-level loss L_{mes} considers the integrity of a candidate. It encourages chord progressions with the same length as the target melody phrases and penalizes the rest. For a given phrase of melody, we consider chord progressions of the same length and we also concatenate shorter chord progressions to make them candidates. Here, we introduce a transition score to penalize the concatenated candidates. We concatenate two progressions r_m^{chord} and r_n^{chord} from the reference template space, and the transition score considers the very two bars connecting them (*i.e.*, the last bar of r_m^{chord} and the first bar of r_n^{chord}). In specific, we count the occurrence of this two-bar chord progression in the dataset and then normalize it by applying a logarithmic function with N, the size of the reference template space

[3] A detailed description of style mapping is provided in our dataset.

as the base. Suppose the occurrence is $c \in \mathbb{N}$, the transition loss of this $m \to n$ concatenated candidate is defined as:

$$T_{m \to n} := 1 + \log_N \frac{1}{c} \tag{2}$$

Moreover, as we do not wish to penalize the non-concatenated candidates, we introduce δ_1 such that $\delta_1 = 1$ if the candidate is concatenated and $\delta_1 = 0$ otherwise. On the other hand, we do not wish to consider candidates with length not equal to the length of the phrase. Hence we introduce δ_2 such that $\delta_2 = 1$ if the candidate has equal length as the melody phrase, and $\delta_2 = e^{-10}$ otherwise. The meso-level loss of the i^{th} phrase is explicitly:

$$L^i_{\text{mes}} := \delta_1 T_{m \to n} + (\frac{1}{\delta_2} - 1) \tag{3}$$

3.2.3 *Macro-level loss*

The macro-level loss L_{mac} computes how well the candidate phrases connect with each other. We consider how smooth the transition is from the previous progression to the current progression using the same transition loss as in Section 3.2.2. For the i^{th} phrase and its corresponding progression candidate, L^i_{mac} denotes the macro-level loss of the i^{th} phrase

$$L^i_{\text{mac}} := \begin{cases} 0 & i = 1 \\ T_{m' \to n'} & i = 2, \cdots, p \end{cases} \tag{4}$$

Here, $T_{m' \to n'}$ is the transition loss defined the same way as in Section 3.2.2. m' and n' index progression candidates to be connected. p is the total number of phrases.

Finally, we define the total loss of the s^{th} candidate of the i^{th} phrase by a weighted sum

$$\begin{aligned} L^{i,s}_{\text{total}} = (\beta(1 - L^{i,s}_{\text{mic}}) + (1-\beta)(1 - L^{i,s}_{\text{mes}})) \\ + \max_t \{L^{i-1,t}_{\text{total}} + \alpha(1 - L^{i,s}_{\text{mac}})\}, \end{aligned} \tag{5}$$

where α, β are parameters that we can tune between 0 and 1. Then we use DP to integrate the three levels of losses and search for the optimal chord progressions which minimize the total loss L^i_{total} at $i = p$.

3.3 An Overview of AccoMontage

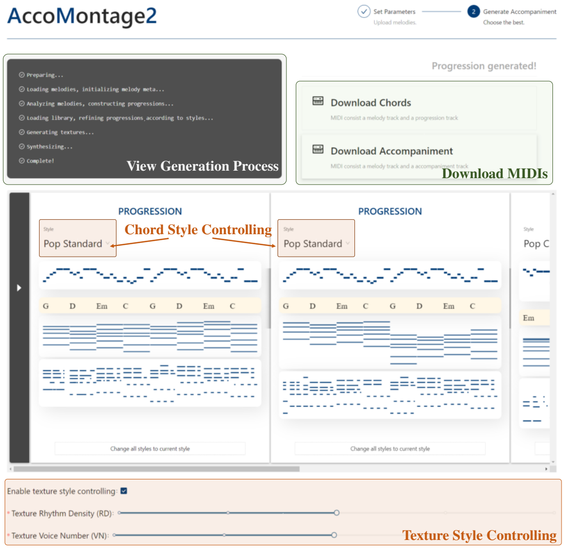
Based on the harmonization results from the last section, we apply AccoMontage [4] to generate complete piano accompaniments in full length. The AccoMontage system offers a hybrid search-style transfer methodology for accompaniment arrangement: given a lead melody together with a chord progression inferred from Section 3.2, it first searches for reference pieces of accompaniments by dynamic programming. It then re-harmonizes the reference accompaniment to the given chord progression by deep learning-based style transfer. Such a pipeline is inspired by common practice that delicate music textures can often be applied in bulk, instead of composing from scratch.

Specifically, reference accompaniment pieces are searched phrase by phrase based on: 1) phrase-level fitness to the melody, and 2) transition smoothness between consecutive phrases. The overall searching process is optimized by the Viterbi algorithm [24]. The re-harmonization is implemented with a VAE framework which is capable of chord-texture disentanglement [1]. By varying the chord representation, we can re-harmonize the accompaniment while keeping its texture. The whole pipeline of AccoMontage secures a delicate accompaniment arrangement with coherent and structured texture. We refer readers to the original work [4] for more technical details.

3.4 Graphical User Interface for Controllability

AccoMontage2 provides a GUI for users to select styles of harmonized chords and accompaniment textures. The process begins with uploading a MIDI file with a melody track and setting the original chord progression styles and piano texture styles. Three labels have to be assigned manually, including phrase boundaries, key, and mode (major or minor in the current version of the system).

The system will then proceed to generation. When the generation is finished, users are able to: 1) listen to and download the generated audio; 2) select a new harmonization style for each individual phrase or the whole; 3) reset the texture style and re-generate the accompaniment. Figure 4 shows the GUI interface of AccoMontage2.



```
Preparing...
Loading melodies, initializing melody meta...
Analyzing melodies, constructing progressions...
Loading library, refining progressions according to styles...
Generating textures...
Synthesizing...
Complete!
```

Progression generated!

Download Chords
MIDI consist a melody track and a progression track

Download Accompaniment
MIDI consist a melody track and a accompaniment track

Style: Pop Standard

Style: Pop Standard

Enable texture style controlling: ☑

Texture Rhythm Density (RD):

Texture Voice Number (VN):

Figure 4: A screenshot of the Graphical User Interface.

4. GENERATION RESULTS

In this section, we showcase one long-term generation result of the AccoMontage2 system. We set $\alpha = 0.1$ and $\beta = 0.5$ in the harmonization algorithm and test our model on a 24-bar melody piece with phrase label `A8A8B8`. The first track in Figure 5 shows the original melody piece and the other two tracks are two different versions of accompaniment generation.

The result shows that AccoMontage2 is able to achieve high controllability in chord style and texture style. On the top of track two (I) and track three (II) in Figure 5, we present using chord notations the harmonization results

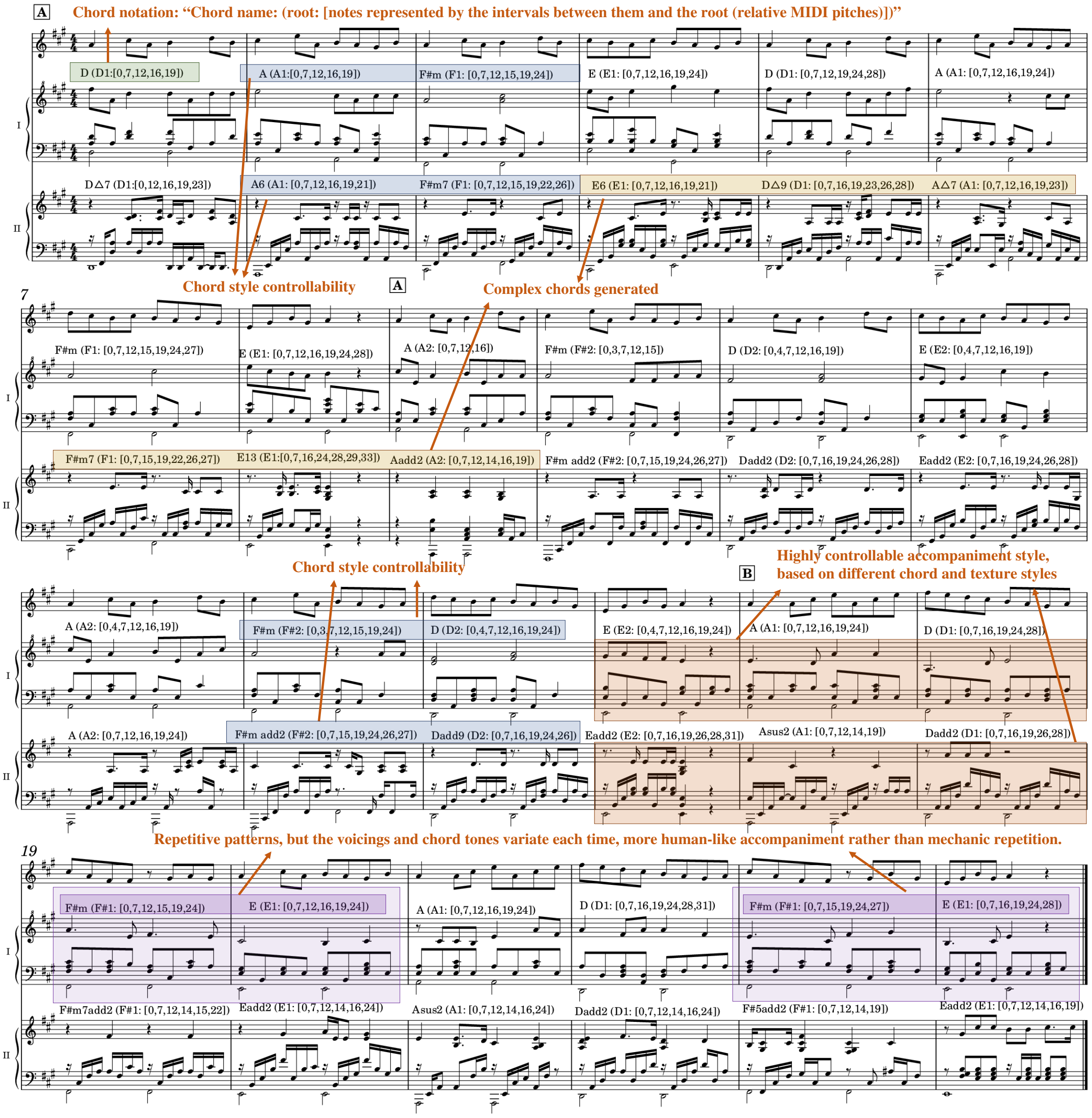

A
Chord notation: "Chord name: (root: [notes represented by the intervals between them and the root (relative MIDI pitches)])"
D (D1:[0,7,12,16,19])
A (A1:[0,7,12,16,19])
F#m (F1: [0,7,12,15,19,24])
E (E1: [0,7,12,16,19,24])
D (D1: [0,7,12,19,24,28])
A (A1: [0,7,12,16,19,24])
I
DΔ7 (D1:[0,12,16,19,23])
A6 (A1: [0,7,12,16,19,21])
F#m7 (F1: [0,7,12,15,19,22,26])
E6 (E1: [0,7,12,16,19,21])
DΔ9 (D1: [0,7,16,19,23,26,28])
AΔ7 (A1: [0,7,12,16,19,23])
II
Chord style controllability
A
Complex chords generated
7
F#m (F1: [0,7,12,15,19,24,27])
E (E1: [0,7,12,16,19,24,28])
A (A2: [0,7,12,16])
F#m (F#2: [0,3,7,12,15])
D (D2: [0,4,7,12,16,19])
E (E2: [0,4,7,12,16,19])
I
F#m7 (F1: [0,7,15,19,22,26,27])
E13 (E1:[0,7,16,24,28,29,33])
Aadd2 (A2: [0,7,12,14,16,19])
F#m add2 (F#2: [0,7,15,19,24,26,27])
Dadd2 (D2: [0,7,16,19,24,26,28])
Eadd2 (E2: [0,7,16,19,24,26,28])
II
Chord style controllability
B
Highly controllable accompaniment style,
based on different chord and texture styles
A (A2: [0,4,7,12,16,19])
F#m (F#2: [0,3,7,12,15,19,24])
D (D2: [0,4,7,12,16,19,24])
E (E2: [0,4,7,12,16,19,24])
A (A1: [0,7,12,16,19,24])
D (D1: [0,7,16,19,24,28])
I
A (A2: [0,7,12,16,19,24])
F#m add2 (F#2: [0,7,15,19,24,26,27])
Dadd9 (D2: [0,7,16,19,24,26])
Eadd2 (E2: [0,7,16,19,26,28,31])
Asus2 (A1: [0,7,12,14,19])
Dadd2 (D1: [0,7,16,19,26,28])
II
Repetitive patterns, but the voicings and chord tones variate each time, more human-like accompaniment rather than mechanic repetition.
19
F#m (F#1: [0,7,12,15,19,24])
E (E1: [0,7,12,16,19,24])
A (A1: [0,7,12,16,19,24])
D (D1: [0,7,16,19,24,28,31])
F#m (F#1: [0,7,15,19,24,27])
E (E1: [0,7,16,19,24,28])
I
F#m7add2 (F#1: [0,7,12,14,15,22])
Eadd2 (E1: [0,7,12,14,16,24])
Asus2 (A1: [0,7,12,14,16,24])
Dadd2 (D1: [0,7,12,14,16,24])
F#5add2 (F#1: [0,7,12,14,19])
Eadd2 (E1: [0,7,12,14,16,19])
II

Figure 5: Harmonization and accompaniment arrangement results for *Dinners 1* from the Nottingham Dataset. The 24-bar melody has an `A8A8B8` phrase structure, and the AccoMontage2 system achieves a high degree of style controllability.

of chord style "Pop-standard" and "Pop-complex" respectively. Track two and track three are the whole accompaniment results. While track two (I) is based on "Pop-standard" and a sparse texture style, track three (II) is based on "Pop-complex" and a dense texture style. We see that both results match with the melody in harmonicity and are able to provide chord and texture variations.

5. EVALUATION

We conduct two comparative experiments to validate our AccoMontage2 system, one for harmonization and the other for accompaniment arrangement. We first show the dataset and the baseline model in Section 5.1 and 5.2. Then we present the experimental results in Section 5.3 and 5.4. Audio examples of our proposed system and ablation studies are available via our GUI link.

5.1 Dataset

For the harmonization experiment, we use Nottingham Dataset [25], which provides around 1000 pairs of the query lead melody and the ground-truth chords. For arrangement generation experiments, we use the POP909 Dataset [26], in which each piece contains a lead melody, annotated chords, and a piano accompaniment. For both data sources, we first select the pieces of meter $\frac{2}{4}$ and $\frac{4}{4}$ and then randomly sample 6 pieces for our experiment. We manually annotate their phrase segmentation and only allow the phrase length of 4 bars and 8 bars.

5.2 Baseline Method

We apply the chord generation model [9] based on bidirectional long short-term memory networks (BLSTM) [27] as our baseline model. This model takes a symbolic melody with the information of time signature, measure, and key as input, and outputs a harmonization result of a sequence of major and minor triads. The BLSTM model considers temporal dependencies by storing both past and future information, reflecting musical context in both forward and backward directions, It is trained on a lead sheet database provided by Wikifonia.org and has achieved reasonable results quantitatively and qualitatively.

5.3 Harmonization Results

We conduct a survey to evaluate the harmonization performance of our model. Our survey has 6 groups of harmonization results and each subject is required to listen to 3 (chosen randomly). Within each group, the subject first listens to a single melody. Each melody is a full-length song randomly selected from the Nottingham Dataset, with an average length of 32 bars (64 seconds). We harmonize the melody using our model and the BLSTM baseline, and additionally acquire an original harmonization using the ground truth chord labels. The subjects are then required to evaluate all three versions of harmonization. The rating is based on a five-point scale from 1 (very poor) to 5 (very high) according to three criteria:

1. **Harmonicity**: How well do the melody and chords stay in harmony with each other;
2. **Creativity**: How creative the harmonization is;
3. **Musicality**: The overall musicality.

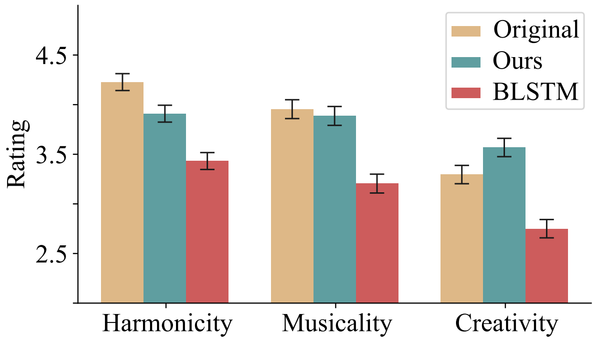


Figure 6: Subjective evaluation for melody harmonization.

A total of 15 subjects with diverse musical backgrounds participated in our survey and we obtain 44 effective ratings for each criterion. As shown in Figure 6, the height of the bars denotes the mean values of the ratings. The error bars stand for the mean square errors (MSEs) computed via within-subject ANOVA [28]. For harmonicity, the original receives the best rating, while our model performs significantly better than the BLSTM baseline. As for the overall musicality, our model is comparable with the original harmonization. For creativity, our model reaches the best. Note that our model supports complex harmonization with voice leading and ninth chords, while the original and the baseline support up to seventh chords in plain root position. Such evaluation results demonstrate that our model introduces *tension* to "flavor" the music while the overall musicality is not affected. For all three criteria, the rating results are statistically significant (p-value $p < 0.05$).

5.4 Accompaniment Generation Results

We conduct another survey to evaluate our model in terms of overall accompaniment generation. In our survey, each subject still listens to 3 groups of generation results (randomly chosen from 6 groups). Within each group, the subjects first listen to a 32-bar melody randomly selected from the POP909 Dataset. Our model generates piano accompaniment through the complete AccoMontage2 pipeline. For the BLSTM baseline, we feed its harmonization results to AccoMontage and obtain the baseline accompaniment. As POP909 contains piano arrangements created by professional musicians for each song, we also have the original accompaniment. The subjects are then required to evaluate all three versions of accompaniment based on the same scale and criteria as in Section 5.3.

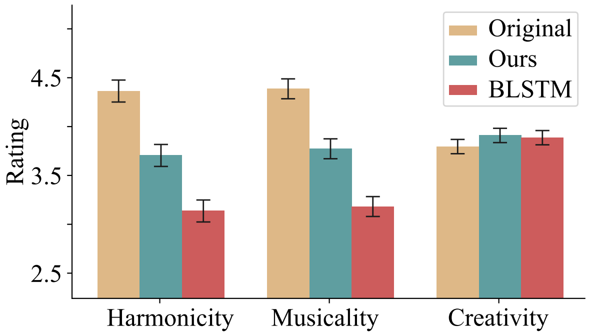


Figure 7: Subjective evaluation for accompaniment arrangement.

We collect a total of 44 effective ratings for each criterion. Figure 7 shows the evaluation results in the same format as in Section 5.3. We report a significantly better performance of our model compared with the BLSTM baseline in harmonicity and musicality ($p < 0.05$), and a marginally better performance than both the baseline and the original in creativity.

6. CONCLUSION AND FUTURE WORK

In conclusion, we contribute a pipeline of algorithms to automatically harmonize and arrange piano accompaniments for melodies of whole-piece popular and folk songs. The system is named after AccoMontage2, built upon its original version which uses a hybrid approach to select accompaniment candidates, edit the candidates using style transfer, and then concatenate them into an organic whole. AccoMontage2 contains two novel modules: a state-of-the-art harmonizer and a GUI for controllable arrangement via interfering in the harmonization and arrangement styles.

In the future, we plan to further optimize the arrangement pipeline by: 1) automatically labeling the melody phrase, 2) extending the model capability to deal with triple meters, and 3) exploring full-band arrangement.

7. REFERENCES

[1] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 662–669.

[2] G. Xia, "Expressive collaborative music performance via machine learning," Ph.D. dissertation, Carnegie Mellon University, USA, 2016. [Online]. Available: https://doi.org/10.1184/r1/6716609.v1

[3] L. Lin, G. Xia, Q. Kong, and J. Jiang, "A unified model for zero-shot music source separation, transcription and synthesis," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 381–388.

[4] J. Zhao and G. Xia, "Accomontage: Accompaniment arrangement via phrase selection and style transfer," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 833–840.

[5] C.-H. Chuan, E. Chew *et al.*, "A hybrid system for automatic generation of style-specific accompaniment," in *Proceedings of the 4th international joint workshop on computational creativity*. Goldsmiths, University of London London, 2007, pp. 57–64.

[6] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 725–734.

[7] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii, "Function- and rhythm-aware melody harmonization based on tree-structured parsing and split-merge sampling of chord sequences," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 502–508.

[8] ——, "Generative statistical models with self-emergent grammar of chord sequences," *Journal of New Music Research*, vol. 47, no. 3, pp. 226–248, 2018.

[9] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 621–627.

[10] Y.-C. Yeh, W.-Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H.-M. Liu, H.-W. Dong, Y. Chen, T. Leong, and Y.-H. Yang, "Automatic melody harmonization with triad chords: A comparative study," *Journal of New Music Research*, vol. 50, no. 1, pp. 37–51, 2021.

[11] C.-E. Sun, Y.-W. Chen, H.-S. Lee, Y.-H. Chen, and H.-M. Wang, "Melody harmonization using orderless nade, chord balancing, and blocked gibbs sampling," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4145–4149.

[12] M. Bretan, G. Weinberg, and L. Heck, "A unit selection methodology for music generation using deep neural networks," *arXiv preprint arXiv:1612.03789*, 2016.

[13] P.-C. Chen, K.-S. Lin, and H. H. Chen, "Automatic accompaniment generation to evoke specific emotion," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[14] Y.-C. Wu and H. H. Chen, "Emotion-flow guided music accompaniment generation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 574–578.

[15] C.-H. Liu and C.-K. Ting, "Polyphonic accompaniment using genetic algorithm with music theory," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–7.

[16] T. Fujishima, "Realtime chord recognition of musical sound: a system using common lisp music," in *Proceedings of the 1999 International Computer Music Conference, ICMC*. Michigan Publishing, 1999.

[17] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[18] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 722–727.

[19] B. Jia, J. Lv, Y. Pu, and X. Yang, "Impromptu accompaniment of pop music using coupled latent variable model with binary regularizer," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[20] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "Popmag: Pop music accompaniment generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1198–1206.

[21] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2837–2846.

[22] N. Kotoulas, "Supersize your midi collection." [Online]. Available: https://www.pianoforproducers.com/nikos-ultimate-midi-pack/

[23] L. Trulla, N. Di Stefano, and A. Giuliani, "Computational approach to musical consonance and dissonance," *Frontiers in Psychology*, vol. 9, p. 381, 04 2018.

[24] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[25] E. Foxley, "Nottingham database," [EB/OL], https://ifdo.ca/~seymour/nottingham/nottingham.html Accessed May 25, 2021.

[26] Z. Wang*, K. Chen*, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, "Pop909: A pop-song dataset for music arrangement generation," in *Proceedings of 21st International Conference on Music Information Retrieval, ISMIR*, 2020.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] H. Scheffe, *The analysis of variance*. John Wiley & Sons, 1999, vol. 72.