KEY AND FUNCTION AWARE MELODY TRIAD HARMONIZATION BASED ON TRANSFORMER MODEL

Huan Zhang[1], *Ran Zhang*[2], *Kun Zhang*[2], *Xiao-rui Wang*[2], *Zhong-yuan Wang*[2]

[1]School of Music, Carnegie Mellon University, Pittsburgh, PA
[2]Kuai Shou Technology Co., Beijing, China

ABSTRACT

In this paper, we describe a transformer-based melody harmonization system, that's aware of the key in melody and harmonic function in chord progressions. We formulate the problem of melody harmonization as a sequence generation problem that's analogical as translation in natural language processing. Thus, we propose utilization of a transformer-based network architecture incorporating neural attention that is able to learn a mapping between melody pitch representations and chord symbol representations. Furthermore, we have proposed a melody harmonization evaluation criteria, that's able to access the harmonization in terms of coherence, rhythm, structure and style. Experiments show that, Our proposed system has outperformed vanilla transformer in various evaluation metrics. From analyzing models trained from three datasets with contrasting music style, we confirm discrepancies exist regarding harmonization of different music genre.

Index Terms— Melody harmonization, Transformer, Deep learning, Music generation

1. INTRODUCTION

Harmony, the vertical arrangement and combination of pitches, plays a crucial role in defining the color and mood of music. The coordination between melody and harmony has long been a problem that interests composers, arrangers and music theorist alike. We present a transformer-based model for automatic melody harmonization, which quantizes music on the time axis and generates a sequence of chord symbols given a sequence of melody notes. Several previous works had attempted to automate the harmonization process [1, 2], and the approaches lies in two general categories: Rule based or probabilistic approaches and deep learning based method.
Traditionally, Hidden Markov Model (HMM) was commonly applied [3, 4] to the harmonization problem, but it suffers from the inability to capture long-term music structures. Genetic Algorithm (GA) has also been employed in four-part harmonization [5], and it's flexible in incorporating human-curated musical rules and conditions. In recent years, music, with its sequential structure and similarity to text, has attracted the adoption of sequential deep learning models [6]. In particular, Transformer models has already proven successful in capturing the long-time structure in the work of Huang et al. [7, 8], generating music with repeating motives and uniform structures. The task of automatic harmonization has also been addressed by Lim et al. [9] with a BiLSTM model, and outperformed HMM-based approaches. Thus, we attempts to extend the deep learning branch by adapting a transformer model, while incorporating key and harmonic function embedding into the melody harmonization task.
Also, despite the existing approaches, few objective evaluation metrics exist for the melody harmonization task. In the work of Yeh et al [10], six metrics were adopted, focusing on the aspect of chord progression and chord-melody harmonicity. However, the structure, rhythm and style of the harmonized chord sequence has yet been addressed.

Paper Contributions. The three main contributions of our work are the following: First, we proposed a Transformer-based model, and demonstrated the effectiveness of adaption of transformer model in music domain such as key and harmonic function embedding and beat sampling representation. Second, we proposed melody harmonization evaluation metrics that access a harmonization from dimensions of consonance, rhythm and variation. Third, we present POP909 Triad Harmonization Dataset, a processed POP909 dataset [11] in fix-length triad lead sheet format. Stylistic differences of harmonization across music genres are compare and contrasted along with two other datasets.

Paper Organization. In section 2, we describe proposed method, model structure, and three datasets being used, preprocessing and representation. In section 3, we document our melody harmonization metrics that access harmonization result. In section 4, we discuss and evaluate the experiment results.

2. METHODOLOGY

2.1. Model Architecture

Figure 1 shows a layer-level visualization of our harmonization network, which follows the encoder-decoder framework

Rendered harmonization samples can be found on drive.google.com/drive/folders/1hqbohzdCerTFjL6mkHG3Z2hhE0mudZnn of transformer model that's originally used for seq2seq language translation. Transformer is an attention-based network that relies on attention mechanism only and does not include recurrent or convolutional architecture. Utilizing multi-head attention together with position-wise fully-connected feed-forward network, it showed significantly faster training speed and achieved better performance than recurrent or convolutional networks for translation tasks. Transformer used scaled dot-product as an attention [12] function:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \tag{1}$$

Our adaptation of the model focuses on the musical aspect, where keys and harmonic functions are incorporated into the embedding.

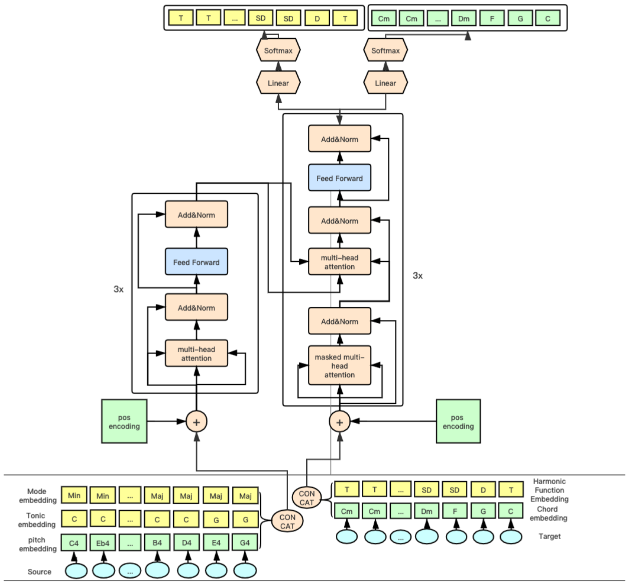

Fig. 1. A layer-level diagram of the harmonization transformer architecture

2.2. Uneven Encoding and beat Sampling

For the quantization of melody and chord sequence, we utilized the uneven tick encoding scheme proposed in [13], which assigns six tokens to a beat. This allows sixteenth notes and triplets to be represented in melody.

However, since passing tones may have less impact on harmony, we also experimented on a quantization of beat sampling sequence, namely only take the pitch on the beat to represent melody and chord sequence (See figure 2). In the example, only pitches on the beat *A-C-F-E* is used as tokens in the sequence, passing tones are disgarded.

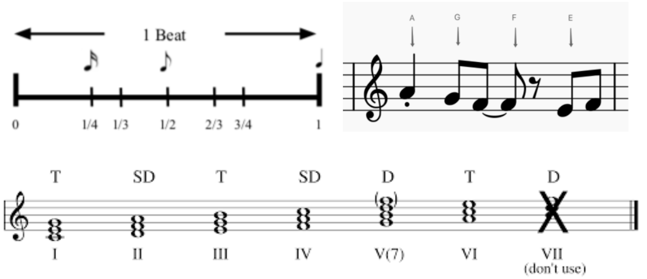

Fig. 2. Left: uneven encoding; Right: beat sampling; Bottom: harmonic function of chords on each degree

2.3. Key Embedding and Harmonic Function Embedding

After concatenating of the one-hot embedding of melody, tonic and mode sequence, the tensor is send into the model as input. For the target sequence, we consider both chords sequence and their harmonic function [14] respect to the key. For the harmonic function sequence, we consider three main harmonic functions: Tonic, Dominant and Subdominant. We define scale degrees 1, 3, 6 as tonic, scale degree 5, 7 as dominant and 2, 4 as subdominant. The chords that doesn't lies within the key scale degrees are labeled as *None*, which separates diatonic and non-diatonic chords.

As shown in 1, the embedding of chords and function are concatenated as target into the encoder. From the decoder output, both predictions for chord sequence and harmonic function sequence are evaluated for cross-entropy loss, and the overall loss is optimized towards their sum.

$$\mathcal{L}(c, \tilde{c}, h, \tilde{h}) = \mathcal{L}_{chord}(c, \tilde{c}) + \mathcal{L}_{func}(h, \tilde{h}) \tag{2}$$

2.4. Data

For the training of our system, we compare and contrast three datasets (Wikifonia, Hooktheory, POP909 [11]), all of which contains melody and chord sequences, with music style span across genres from folk, jazz to pop-songs.

In order to standardize the datasets, we limit our chordal vocabulary into triads, namely chords with *Major*, *Minor*, *Diminished* and *Augmented* quality. The melody pitch vocabulary is limited into the range from *C3* to *C6*, and data is transposed (moving a collection of pitches up or down a constant interval) to fit into this range. Also, we limit our rhythm choice by music in $\frac{4}{4}$ time. The data that doesn't satisfy these constraints are either simplified or pruned.

Our pitch representation (both in melody and chordal root) separates enharmonic equivalences, namely that $C\sharp$ and $B\flat$ are taken as different classes. This is to be coherent with the key embeddings, as only pitch $C\sharp$ exist within the key of $C\sharp$, not pitch $B\flat$, even they are enharmonic in terms of frequency.

Since key modulation (key changes during music) happens frequently in the data, we want our model to be aware of the key color when harmonizing melody. Thus, we extract tonic (The first scale degree of the key) sequence and mode (*Major*, *Minor*) sequence that are quantized in the way as melody.

For data augmentation, we transpose the piece by 12 intervals within an octave in order to obtain data evenly distributed in all keys. Statistics of 3 datasets we used are shown in Table 1. Also, to standardize the experiment, we truncate the music with a maximum length of 60 measures, and removed the trailing empty measures (measures that doesn't have both melody and harmony). Note that we does not make any assumption on the structure of music data.

	Count	Avg Length	Style
Wikifonia	4536	49.8	Folk, Jazz, Pop...
Hooktheory	11360	9.8	Pop, Rock...
POP909	909	74.5	Chinese pop

Table 1. Dataset statistics

3. HARMONIZATION EVALUATION METRIC

A piece of melody could have multiple ways of musically-meaningful harmonization, thus only considering the accuracy with ground truth chord sequences itself does not suffice as evaluation. The metrics we proposed to evaluate the melody harmonization task addresses consonance, rhythm and variation. Under these categories we obtain seven metrics scores.

Interval Coherence (IC): We adopt the same definition of Interval Coherence with [10], where it measures the consonance of the interval between melody pitch and each of the chord pitch.

Pitch Class Entropy (PE): We use the pitch class information entropy to capture the notion of vertical coherence. Among each chordal window (the span of duration of each chord), we weight the probability of each pitch class by how long it appears. The information entropy is calculated across 12 pitch bins. Then the vertical coherence of the whole piece is obtained by averaging entropies of all chordal windows, with weighting the window duration.

$$PE = \frac{\sum_{w \in windows} \sum_{i=0}^{11} P(x_i) log(P(x_i)) \times dur(w)}{|windows|}$$

Harmony Variation (HV): We also expect more unconventional harmonies instead of similar chord progressions each time. Based on the count of each chord in the overall dataset, we capture the harmony variation of a piece by the sum of inverse frequency of the chord in the dataset corpus, weighted by the duration of each chord.

Tonality Enhancement (TE): In a consonant harmonization, we expect the notes in the harmony enhances the key of melody, or at least don't contradict it. For the original melody pitches and overall harmonized pitches (chordal pitches plus melody pitches), 24 key estimation probability is computed by Krumhansl-Schmuckler key estimation algorithm [15]. Between the two key estimation distributions, a correlation coefficient is computed as key coherence score.

Repetition Score (RS): When harmonizing, the model should be aware of the musical structure implied by melody. As structural analysis is not trivial [16, 17], we simplify the notion of melody structure into repetition: For each pair of the window of one measure, if the melody repeats, we calculate the portion of overlap in their chordal accompaniment. The repetition overlap scores of all repeating melody window are averaged.

Harmonic Rhythm Standard Deviation (HS) and Mean (HM): When harmonizing, the chord sequence implicitly creates a rhythmical structure, known as harmonic rhythm. Music of different genre and style embodies different trends for harmonic rhythm. To capture such trends, we take the duration of each chord in the generated sequence, and compute their standard deviation, where a lower std represents a more stable harmonic rhythm. Harmonic rhythm mean indicates roughly for what duration does the chord occupy before a chord change.

4. RESULT ANALYSIS AND DISCUSSION

4.1. Architecture Comparative Results

For the comparative analysis of model structure, we split the Wikifonia dataset into training, validation and testing data in 8:1:1 ratio, and all models were trained for 1000 epochs, using the Adam optimizer [18] with a learning rate of 0.001, and batch size of 96. For the comparative analysis, we applied the harmonization evaluation metrics 3 on each of the following architectures: (1). Proposed architecture (**S**): Key embedding and Harmonic Function embedding with transformer architecture that optimized towards the summed loss 2. (2). Proposed architecture without beat sampling (**B**) in 2.2, that contains full sequences. (3). Baseline vanilla transformer (**V**), with only melody to harmony sequence as input and output.

The results of comparative experiment are shown in Figure 4, where we compares the architectures with the original harmonization in Wikifonia dataset. For interval coherence and tonality enhancement, we can see that adopting extra input and output embedding yields a significant boost on the result compared to the baseline. This demonstrates the effectiveness of key and harmonic function information in predicting a more coherent harmonization. Also, without the harmonic function which serves as a higher level representation of the harmony, the predicted chord sequence contains much more repetition and less variation. Finally, we note that the

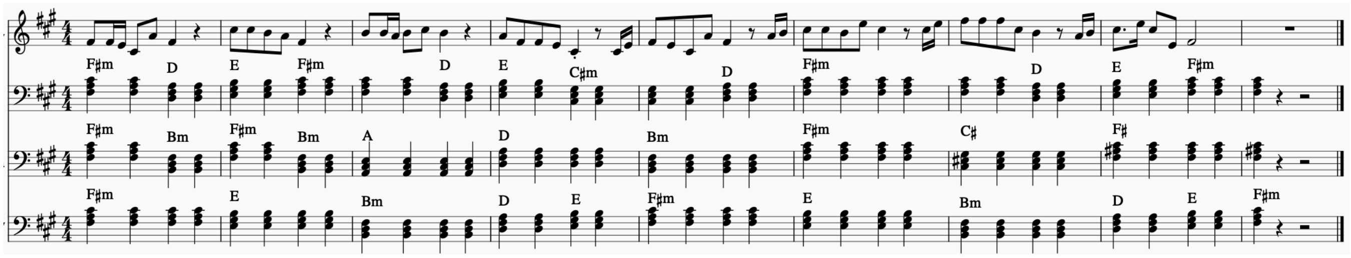

Fig. 3. Three different harmonizations of Chinese pop song *Drunken Butterfly*, inferred from model trained with different datasets. From top to bottom: (a) POP909; (b) Wikifonia; (c) Hooktheory

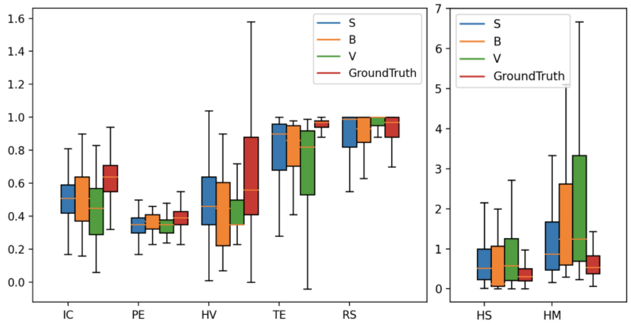

Fig. 4. Metrics for all of the tested architectures on Wikifonia dataset

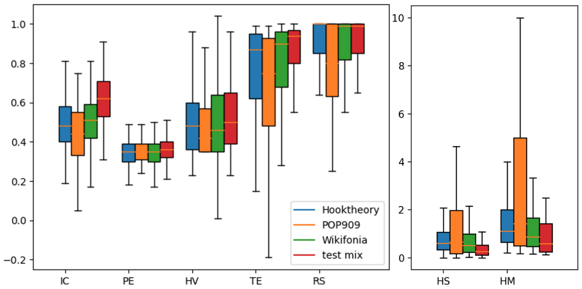

Fig. 5. Harmonization results compared between models trained on three different datasets.

beat sampling representation yielded a result similar to the unsampled representation, demonstrating that the tones not on the musical beats indeed plays a minor role in the harmonization task. However, in terms of harmonic rhythm, there is still a large discrepancy between tested architectures and original harmonization.

4.2. Dataset and Style Analysis

In order to investigate the stylistic differences that emerged from music across different genre, we trained individual models with proposed architecture from three datasets described in Section 2.4. For the evaluation, 80 piece from the testing split of each dataset is randomly chosen to form a mixed set, and each of the three models inferenced harmonization on this mixed set. Their evaluation result, as well as the benchmark of the original harmonization from test data, are shown in Figure 5.

Several interesting trends can be observed: 1. **Style generalization**: None of the model is able to fully generalize across the mixed style, as the original harmonization scores higher on all metrics. 2. **Rhythmic Coherence**: On the mix test set, all tested models achieves a harmonic rhythm of roughly one chord change per measure. 3. **Tonality bias**: The Tonality Enhancement ability learned from POP909 dataset is significantly lower than the other two. This might be a result of the use of traditional modes like pentatonic scale in Chinese music.

Figure 3 illustrates how different harmonization styles learned by the model reflects on same piece of music. From the style of Wikifonia which is biased towards folk, jazz and classical music, a *Picardy Third* (C♯ - F♯) is enforced on this Chinese folk tune, ending the tune by a cadence into major key, which sounds quite out-of-context. In comparison, the more diverse Hooktheory dataset and more Chinese-oriented POP909 datasets yields more stylistic coherent result with this tune. Also, in terms of harmonic rhythm, the style from Hooktheory segments tends to be more stable (õne chord per measure), and the style from POP909 captures more of short music phrase like the half cadence in the example.

5. CONCLUSION

In this paper, we proposed a transformer-based melody harmonzation model that's able to generate chord sequence that's musically coherent with the given melody. With the key and harmonic function embedding, our model outperforms the baseline. A comparative study between different datasets is performed and confirms the discrepancy between music style and genre when harmonizing melodies.

6. REFERENCES

[1] Ching-Hua Chuan and Elaine Chew, "A hybrid system for automatic generation of style-specific accompaniment," in *Proc. int. joint workshop on computational creativity*, 01 2007.

[2] Dimos Makris, Ioannis Karydis, and Spyros Sioutas, *Automatic Melodic Harmonization: An overview, challenges and future directions*, 06 2016.

[3] Jean-François Paiement, Douglas Eck, and Samy Bengio, "Probabilistic melodic harmonization," in *Advances in Artificial Intelligence*, Luc Lamontagne and Mario Marchand, Eds., Berlin, Heidelberg, 2006, pp. 218–229, Springer Berlin Heidelberg.

[4] Stanisław A. Raczyński, Satoru Fukayama, and Emmanuel Vincent, "Melody harmonization with interpolated probabilistic models," *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, 2013.

[5] Somnuk Phon-Amnuaisuk and Geraint Wiggins, "The four-part harmonisation problem: A comparison between genetic algorithms and a rule-based system," *Proceedings of the AISB'99 Symposium on Musical Creativity*, 08 2001.

[6] Jean-Pierre Briot, Gaëtan Hadjeres, and Francois Pachet, *Deep Learning Techniques for Music Generation - A Survey*, 09 2017.

[7] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music transformer," in *International Conference on Learning Representations*, 2019.

[8] Yu-Siang Huang and Yi-Hsuan Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, New York, NY, USA, 2020, MM '20, p. 1180–1188, Association for Computing Machinery.

[9] Hyungui Lim, Seungyeon Rhyu, and Kyogu Lee, "Chord generation from symbolic melody using blstm networks," 10 2017.

[10] Yin-Cheng Yeh, Wen-Yi Hsiao, Satoru Fukayama, Tetsuro Kitahara, Benjamin Genchel, Hao-Min Liu, Hao-Wen Dong, Yian Chen, Terence Leong, and Yi-Hsuan Yang, "Automatic melody harmonization with triad chords: A comparative study," 2020.

[11] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia, "Pop909: A pop-song dataset for music arrangement generation," 08 2020.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., pp. 5998–6008. Curran Associates, Inc., 2017.

[13] Ashis Pati, Alexander Lerch, and Gaëtan Hadjeres, "Learning to traverse latent spaces for musical score inpainting," 11 2019.

[14] Nori Jacoby, N. Tishby, and Dmitri Tymoczko, "An information theoretic approach to chord categorization and functional harmony," *Journal of New Music Research*, vol. 44, pp. 219–244, 2015.

[15] Søren Madsen and Gerhard Widmer, "Key-finding with interval profiles," *International Computer Music Conference, ICMC 2007*, 01 2007.

[16] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck, "A hierarchical latent vector model for learning long-term structure in music," 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 4364–4373.

[17] Shuqi Dai, Huan Zhang, and Roger Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," in *Proceedings of the 2020 Joint Conference on AI Music-Creativity*, 2020.

[18] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.