Discovering Music Relations with Sequential Attention

Junyan Jiang[1], **Gus G. Xia**[1], **Taylor Berg-Kirkpatrick**[2]
[1]Music X Lab, NYU Shanghai
[2]Department of Computer Science and Engineering, University of California San Diego
`{jj2731,gxia}@nyu.edu, tberg@eng.ucsd.edu`

Abstract

The element-wise attention mechanism has been widely used in modern sequence models for text and music. The original attention mechanism focuses on token-level similarity to determine the attention weights. However, these models have difficulty capturing sequence-level relations in music, including repetition, retrograde, and sequences. In this paper, we introduce a new attention module called the sequential attention (SeqAttn), which calculates attention weights based on the similarity between pairs of sub-sequences rather than individual tokens. We show that the module is more powerful at capturing sequence-level music relations than the original design. The module shows potential in both music relation discovery and music generation.[1]

1 Introduction

Music is one type of sequential data with distinctive structures. Various kinds of similarity occur among different phrases of a single music piece. Many music relations are based on sequence-level similarity. For example, a modulated sequence describes a music relation where two phrases' rhythm is the same, but the pitches are shifted.

A well-known method to capture relations in a sequence is the transformer model (Vaswani et al., 2017). Transformer-based models have had recent success in sequence generation and representation learning for both text (Radford et al., 2019; Devlin et al., 2018) and music (Huang et al., 2018; Dhariwal et al., 2020).

The core mechanism of the transformer is the element-wise attention layer. The attention module allows information exchange between any tokens in the sequences. However, it is not an explicit inductive bias for direct sequence-to-sequence matching. Second, a multi-layer attention setting is required: the model needs to collect the sequential information using the positional embedding (Vaswani et al., 2017; Shaw et al., 2018) on the first layer, and then compare the sequential information on the subsequent layers. These problems make the model hard to train and require additional parameters, which may also harm the model's generalization ability.

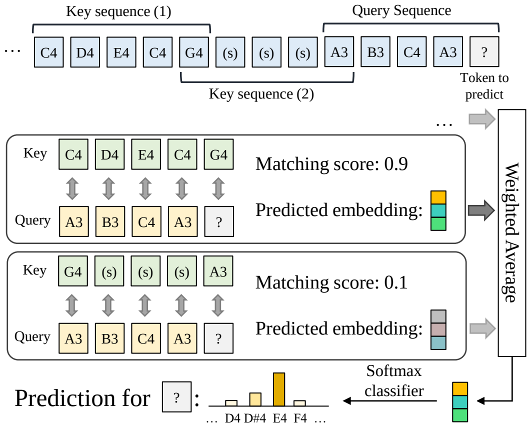

Figure 1: An overview of a self-attentive monophonic language model with the sequential attention mechanism. The model tries to predict the next token (represented by a question mark) by attending to related subsequences appear previously. (1) and (2) show two potential alignments. The model assigns a larger weight (matching score) to key sequence (1) over (2) since key sequence (1) has strong relations (tonal sequence) with the query sequence and can help to predict the next token (E4 in this case).

In this paper, we propose the sequential attention module, a new attention module that explicitly models sequence-level music relations. In this module, we measure the similarity of two sequences by a token-wise comparison instead of the dynamic time warping approach (Walder and

[1] Code and pre-trained models are available at `https://github.com/music-x-lab/SeqAttn`

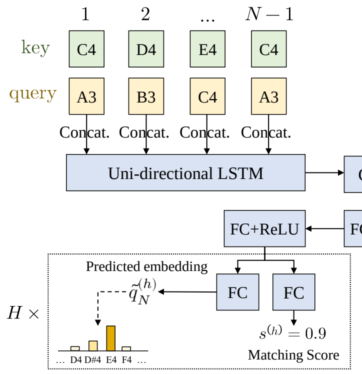

Figure 2: The architecture of the sequential attention unit with H heads. The model takes in the key and the query sequence and outputs the matching scores $s^{(h)}$ and the predicted embedding $\tilde{q}_N^{(h)}$ for each head.

Kim, 2018; Hu et al., 2003) to ensure time efficiency. We also show how to build a self-attentive language model based on the module to capture phrase-level self-similarity in a music piece. An overview of the process is shown in Figure 1. We show by experiments that the proposed model is better at capturing such self-similarity than the transformer model with a comparable size.

2 Proposed Method

2.1 Sequential Attention Unit

We first introduce the basic unit of the proposed module. The design is shown in Figure 2. Assume that we have two sequences of equal length, the query sequence $\mathbf{q}=(q_1,q_2,...,q_N)$ and the key sequence $\mathbf{k}=(k_1,k_2,...,k_N)$. Each q_n, k_n is an embedding vector of dimension d_v^s. Here, q_N is unknown while $\mathbf{k}_{1...N}$ and $\mathbf{q}_{1...N-1}$ are known. The target of the unit is to (1) estimate their matching score (s) between $\mathbf{q}$ and $\mathbf{k}$, and (2) if they are well matched, predict the unknown element q_N given the corresponding key element k_N.

The module uses a multi-head setting (Vaswani et al., 2017) to allow learning multiple distinct relations between the same $\mathbf{q}$, $\mathbf{k}$ pair. For a sequential attention unit with H attention heads, we have:

$$[s^{(1...H)};\tilde{q}_N^{(1...H)}]=\mathrm{SeqAttn}(\mathbf{q}_{1...N-1},\mathbf{k}_{1...N},e) \tag{1}$$

where e is a relative positional embedding vector. We first concatenate the corresponding elements in the query and key sequences, as well as the relative positional embedding ($f_n=[q_n;k_n;e]$), and feed them to a uni-directional LSTM. The last hidden state h_n and the last key element k_n are used to estimate the matching score $s^{(h)}$ and the predicted $\tilde{q}_N^{(h)}$ for each head $h=1...H$:

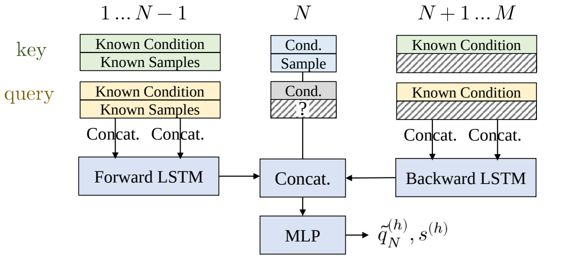

Figure 3: The architecture of the conditional sequential attention unit with H heads. The Multi-Layer Perceptron (MLP) has the same architecture as the unconditioned module.

$$h_N=\mathrm{LSTM}(f_1,f_2,...,f_{N-1}) \tag{2}$$

$$[s^{(1...H)};\tilde{q}_N^{(1...H)}]=\mathrm{MLP}([h_N;k_N]) \tag{3}$$

where MLP is a multi-layer perceptron with 3 fully connected layers and Rectified Linear Unit (ReLU) activations.

2.2 Self-Attention Layer

We now consider how to integrate the module into a language model using self-attention. Self-attention is a method to generate new tokens in a sequence by attending to previously generated ones. Given a partial sequence $x_{1...N-1}$, we want to predict x_N. We first enumerate the distance $i=1,2,3,...$ between the query sequence and the key sequence. For each i, we calculate the matching score s_i and the predicted embedding $\tilde{x}_{N,i}$:

$$[s_i^{(1..H)};\tilde{x}_{N,i}^{(1..H)}]=\mathrm{SeqAttn}(\mathbf{x}_{1...N-1},\mathbf{x}_{1-i...N-i},e_i) \tag{4}$$

where e_i is a learned relative positional embedding for distance i. We will assign $x_k=\mathbf{0}$ for all non-positive indices $k\leq 0$. Then, a weighted average of $\tilde{x}_{N,i}$ is calculated as a final prediction. For each head $h=1...H$, we have:

$$\hat{s}_i^{(h)}=\frac{\exp(s_i^{(h)})}{\sum_{i'}\exp(s_{i'}^{(h)})} \tag{5}$$

$$\tilde{x}_N^{(h)}=\sum_i \hat{s}_i^{(h)}\tilde{x}_{N,i}^{(h)} \tag{6}$$

$$\tilde{x}_N=\mathrm{Linear}([\tilde{x}_N^{(1)};...;\tilde{x}_N^{(H)}]) \tag{7}$$

We can then use $\mathrm{Softmax}(\mathrm{Linear}(\tilde{x}_N))$ to predict the probability of the actual tokens for x_n.

In practice, we do not enumerate all i values since most of the alignments do not agree with the rhythmic structure, thus less meaningful to perform the comparison. We can eliminate such cases to make the model more efficient. See section 3.2 for a detailed setting.

2.3 Conditional Sequential Attention

For the conditional sequence generation task, we propose the modified sequence attention unit, as shown in Figure 3. Here, we want to generate a target sequence x^s given a known condition sequence x^c (e.g., to generate the melody given the chord sequence). The major modification is that we add a backward LSTM to match the future conditions in order to generate the current token.

Assume we have the query sequence $\mathbf{q} = (q_1, q_2, ..., q_M)$ and the key sequence $\mathbf{k} = (k_1, k_2, ..., k_M)$ of equal length M. Each $q_n = (q_n^c, q_n^s)$ and $k_n = (k_n^c, k_n^s)$ are now a tuple of the sample and the condition. We assume that all conditions $k_{1..M}^c$, $q_{1..M}^c$ and a part of the samples $k_{1..N}^s$, $q_{1..N-1}^s$ are known ($N \leq M$). We are interested in estimating q_N^s. In this case, we change the Eqn. 2 and 3 to the following:

$$\overrightarrow{h}_N = \text{LSTM}_{\text{fw}}(f_1, f_2, ..., f_{N-1}) \tag{8}$$

$$\overleftarrow{h}_N = \text{LSTM}_{\text{bw}}(b_M, b_{M-1}, ..., b_{N+1}) \tag{9}$$

$$[s^{(1...H)}; \tilde{q}_N^{(1...H)}] = \text{MLP}([\overrightarrow{h}_N; \overleftarrow{h}_N; k_N; q_N^c]) \tag{10}$$

where $f_n = [k_n; q_n; e]$ and $b_n = [k_n^c; q_n^c; e]$. The forward LSTM tries to match the previous samples and the conditions, while the backward LSTM tries to match the future conditions only.

3 Experiments

3.1 Dataset

We trained and evaluated the proposed method on two datasets of different genres: (1) the Nottingham dataset (Foxley, 2011), an American folk dataset with 1,021 songs after filtering; (2) the POP dataset, a privately collected dataset with 1,394 Chinese pop songs with a 4/4 meter. All songs have a monophonic melody line with chord labels. For each dataset, we use 80% songs for training, 10% for validation, and 10% for testing. We augment the training set by pitch-shifting within the range [-5,6] semitones.

We quantize all songs to a sixteenth-note level. We represent each melody token as one of the 130 states: 128 onset states (for the 0-127 MIDI pitch range), 1 *sustain* state and 1 *silence* state. Each chord is encoded into a 36-dimensional multi-hot vector: 12 dimensions for the root scale, 12 dimensions for the bass scale, and 12 dimensions for its pitch classes.

Model	Nottingham		POP	
	Acc.	Ppl.	Acc.	Ppl.
Unconditioned models				
Mode	61.04	-	52.26	-
Ours+BA	**88.23**	**1.54**	**84.08**	**1.77**
Ours+MA	-	-	79.24	2.09
Transformer	84.58	1.70	70.69	2.73
Chord-conditioned models				
Ours+BA	**90.26**	**1.40**	**85.27**	**1.68**
Ours+MA	-	-	82.44	1.88
Transformer	84.87	1.66	71.30	2.61

Table 1: The comparative results for the accuracy and the perplexity of next token prediction on test sets.

3.2 Model Training

We implement both the conditional and unconditional models using sequential attention with $H = 4$ attention heads. We use $d_v^s = 256$ as the note embedding dimension, $d_v^c = 128$ as the chord embedding dimension, and $d_{\text{hidden}} = 256$ as the hidden dimension of the LSTM and MLP layers.

As mentioned in section 2.2, we only select the distance values i that leads to rhythmic meaningful alignments:

$$i \in \{i \in \mathbb{Z} | k \bmod i = 0 \text{ or } i \bmod k = 0\} \tag{11}$$

where k is a pre-defined group size. We experimented on two different selections: $k = 4$ for beat-level alignment (BA) and $k = 16$ for measure-level alignment (MA, only for 4/4 meter songs). For the Nottingham dataset, we only use beat-level alignment since it contains meter changes.

We define the model loss as the cross-entropy loss for the next token prediction task. The model is trained using the Adam optimizer (Kingma and Ba, 2014) with a constant learning rate of 1e-4. The training is stopped when the validation loss is not improved in 20 epochs.

To increase the robustness of the model performance, we randomly drop key-value pairs with a probability of 50% during training to encourage the model to discover more relations in a piece. The attention dropout is not used in testing.

3.3 Comparative Results

We first compare the proposed method against baseline methods for the next token prediction task. To predict the next token in a partial phrase, it is beneficial if the model learns to attend to sim-

	Input Sequence	Prediction	Ref.
(1)	A4 (s) B4 (s) C5 (s) G4 (s) F4 (s) (s) (s) E4 (s) (s) (s) A4 (s) B4 (s) C5 (s) G4 (s) F4 (s) (s) (s) ?	E4: 89.40% D4: 2.20%	E4
(2)	G4 (s) A4 (s) G4 (s) F4 (s) E4 (s) D4 (s) C4 (s) (s) (s) F4 (s) G4 (s) F4 (s) E4 (s) D4 (s) C4 (s) ?	Bb3: 51.24% B3: 36.85%	B3
(3)	C4 (s) D4 (s) E4 (s) F4 (s) G4 (s) E4 (s) C4 (s) G3 (s) D4 (s) E4 (s) F#4 (s) G4 (s) A4 (s) F#4 (s) D4 (s) ?	A3: 21.85% (s): 14.68%	A3

Table 2: A case study of the module's behavior for different music relations: (1) exact repetition, (2) tonal sequence and (3) modulating sequence. The question mark is the token to predict and the (s) token is the *sustain* label. The table shows the top two predictions and their probability from the sequential attention model.

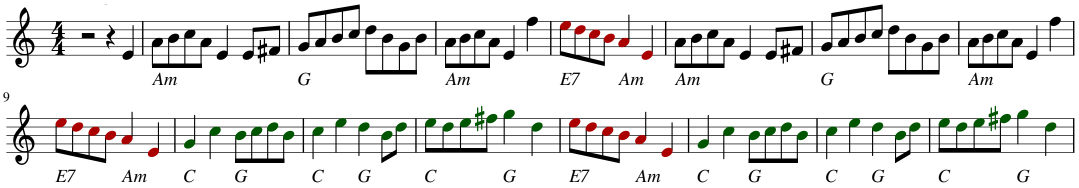

Figure 4: A generated sample. All chords and the melody for the first 8 bars are given. The model generates the melody for the next 8 bars. The repetitions in the generated piece are painted in colors (green and red).

ilar phrases appear previously. We use two baseline methods: (1) a weak baseline (Mode) that always predicts the most frequent token (the *sustain* label), and (2) a 3-layer transformer model with relative positional embedding (Shaw et al., 2018). The model has a transformer width of 256 and 4 attention heads. The results are listed in table 1. Results show that our proposed method acquires higher accuracy and lower perplexity on both the unconditioned model and the chord-conditioned model.

3.4 Analysis of the Attention Module

To further investigate the types of music relations that the sequential attention module captures, we apply the unconditional model with measure-level alignment to three 2-bar test cases with different music relations: (1) exact repetitions (2) tonal sequences and (3) modulating sequences, as shown in table 2. The model predicts reasonable results for all three test cases. Notice that the top 2 predictions of case (2) both form valid tonal sequences (in C major and F major keys, respectively). The model learns such music relations through self-supervision without explicit human instructions.

3.5 Music Generation

We also perform a music generation task using the conditioned language model. Figure 4 shows a generated example where we generate the next 8 bars of melody according to the chords and the first 8 bars of the melody of a sample (reelsd-g18.mid) from the Nottingham test set. In this example, the model learns to repeat the phrases with the same chord sequences and to very if the chord sequences changes.

However, as the model only performs token-by-token prediction, it lacks control over the global music structure. We found some generated examples have too many repetitions or too early cadences. Generating music with controlled music structures are left as a future work.

4 Conclusion

In this paper, we propose a new attention module, the sequential attention module, that explicitly models similarity between two sequences. Based on the module, we implement a self-attentive music language model. The model discovers and captures the self-similarity in music pieces and improves the next token prediction results.

Several important tasks are left as future works. First, the proposed method cannot capture music relations of different time scales since the sequential attention module performs a token-wise alignment of the query and the key sequence. A different module design is required in this case. Second, we want to explore whether the discovered relations can help us in other analysis and generation tasks, e.g., automatic music segmentation, and automatic music accompaniment.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

E. Foxley. 2011. Nottingham database.

Ning Hu, Roger B Dannenberg, and George Tzanetakis. 2003. Polyphonic audio matching and alignment for music retrieval. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No. 03TH8684)*, pages 185–188. IEEE.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music transformer. *arXiv preprint arXiv:1809.04281*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Christian J Walder and Dongwoo Kim. 2018. Neural dynamic programming for musical self similarity. *arXiv preprint arXiv:1802.03144*.