Melody Infilling with User-Provided Structural Context

reurl.cc/eOjGlK

Check the demo website!

Chih-Pin Tan[1,2] Alvin W.Y. Su[1] Yi-Hsuan Yang[2]

[1]National Cheng Kung University [2]Academia Sinica

Abstract

Music infilling *without content controlled* has been well done by recent researches:

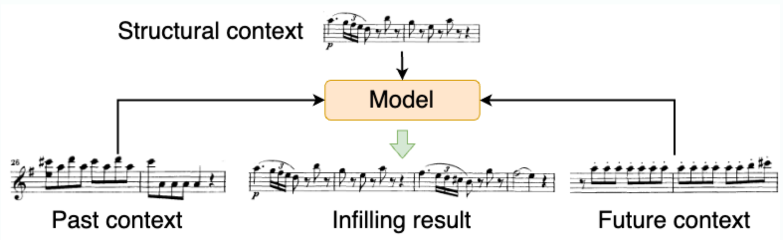
➡ Help composers create music by their outline? No

We propose a Transformer-based model for music infilling:

➡ Accept past context, future context, and **extra structural contexts** as the model input

➡ Use special segment embedding to generate missing part **before** future context

➡ Generate result with structural correspondence to provided-context



Dataset

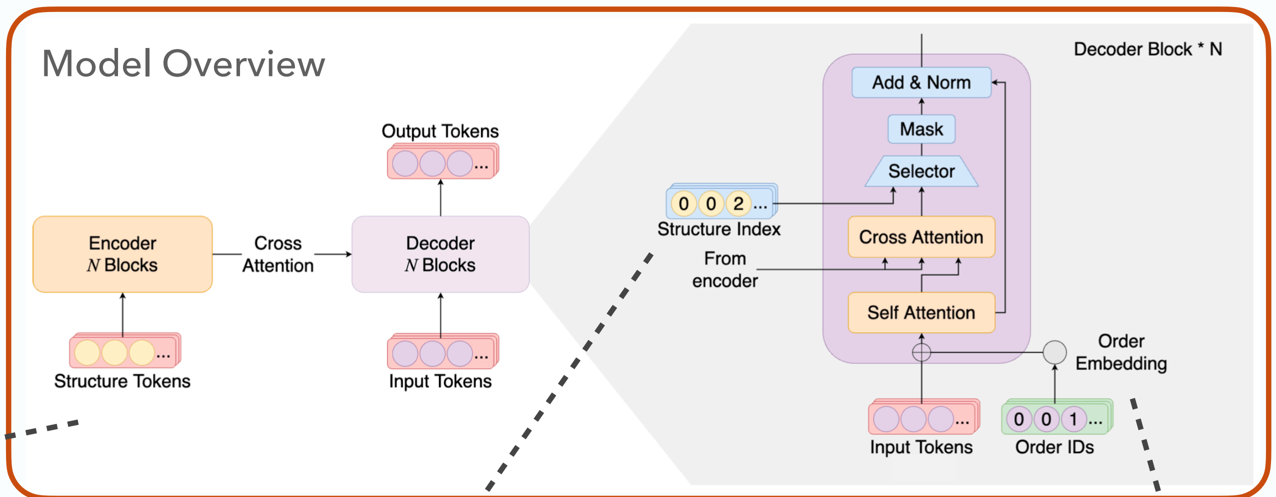
POP909 MIDI songs with structure labels:

- 3 tracks: melody, bridge, and piano
  - ➡ We remove piano to simplify the data
- Similar phrases (often having structural correspondence in pop songs) are marked with the same letters:
  - ➡ Ex: ***A4B4A4B4*** means bars 1~4 are similar to bars 9~12, and bars 5~8 are similar to bars 13~16. Phrases belong to **A** sound different from phrases belong to **B**.

Architecture

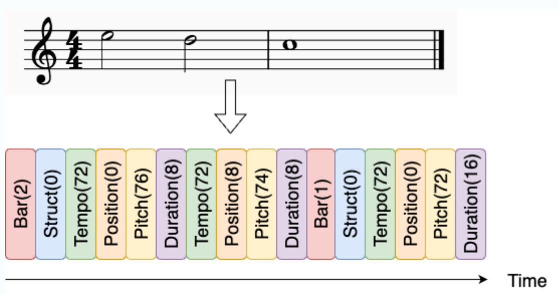
Transformer-based Model:

- Carry structural information & control length:
  - ➡ REMI-based token representation
- Generate middle part:
  - ➡ Order embedding (segment embedding)
- Refer to corresponding context:
  - ➡ Attention-selecting



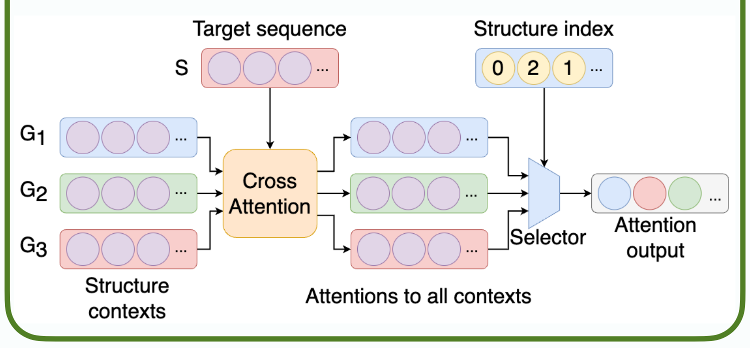
REMI-based Representation

We Convert music into tokens with 6 kinds of event: *Tempo*, *Position*, *Pitch*, *Duration*, *Bar*, and *Struct*. The number assigned to ***Bar*** indicates the remaining length of generated contents. ***Struct*** is the reference of corresponding structural context.



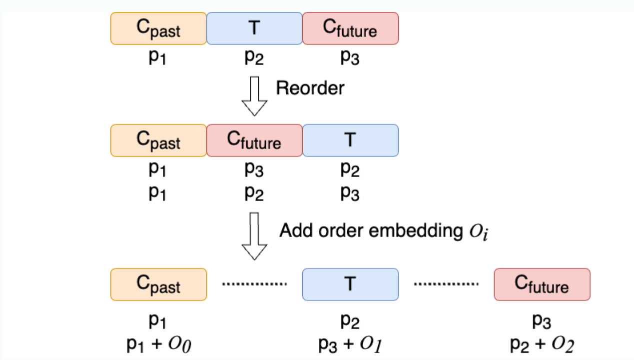
Attention-Selection

The encoder processes multiple sections of the structural context at once, where each section corresponds to a unique ***Struct*** token. The decoder select the right structural context for the generated content on cross-attention.



Order Embedding

Vanilla Transformer cannot generate middle part due to the limitation of sequential generative model. We **reorder** the target to the last position of the prompt and add order embedding to let the model know the target **sitting on the middle** actually.



Experiment

Objective Evaluation

There are 3 objective metrics used:

- Pitch Class Histogram Cross Entropy (***H***)
  - ➡ Cross entropy of note pitches
- Grooving Pattern Similarity (***GS***)
  - ➡ Rhythmic similarity of onset patterns
- Melody Distance (***D***)
  - ➡ Similarity of two melody lines

The values of ***H*** and ***GS*** represent that each model performs well on connecting prompts. The lower value of ***D*** shows that our model behaves better on the structural correspondence.

	$H \downarrow$	$GS \uparrow$	$D \downarrow$
Ours	**2.75**±0.80	**0.70**±0.08	**25.73**±19.45
VLI	3.47±1.57	0.67±0.09	49.40±25.12
Hsu	9.87±4.64	0.64±0.09	65.41±38.00
Original	2.78±0.89	0.70±0.09	0.00± 0.00

User Study

Subjects are asked to evaluate the result with 4 aspects:

- Melody fluency (**M**)
- Rhythmic fluency (**R**)
- Structural corresponded (**S**)
- Overall (**O**)

Our model get higher score in each aspects.

		M	R	S	O
all	Ours	**3.46**	**3.51**	**3.40**	**3.42**
	VLI	2.96	3.14	3.12	2.97
	Hsu	2.60	2.95	2.75	2.64
	Real	**3.77**	**3.77**	**3.62**	**3.66**
pro	Ours	**3.58**	**3.28**	**3.28**	**3.42**
	VLI	2.67	2.86	2.78	2.72
	Hsu	2.36	2.75	2.39	2.44
	Real	**3.61**	**3.56**	**3.42**	**3.42**

Over-Imitation

Model generates the result by directly copying the structural context

- Solution (after formal submission)
  - ➡ Consider the loss of past context on training
- Check it on the demo page and Github!

Conclusion

- Teach the model how to generate content by following the user's guidance
- Another way to provided the context instead the whole music segment