International Conference on New Interfaces for Musical Expression

COSMIC: A Conversational Interface for Human-AI Music Co-Creation

Yixiao Zhang[1], Gus Xia[2], Mark Levy[3], Simon Dixon[1]

[1]Center for Digital Music, Queen Mary University of London, [2]NYU Shanghai, [3]Apple Inc.

License: Creative Commons Attribution 4.0 International License (CC-BY 4.0)

ABSTRACT

In this paper, we propose COSMIC, a **CO**nver**S**ational Interface for Human-AI **M**us**I**c **C**o-Creation. It is a chatbot with a two-fold design philosophy: to understand human creative intent and to help humans in their creation. The core Natural Language Processing (NLP) module is responsible for three functions: 1) understanding human needs in chat, 2) cross-modal interaction between natural language understanding and music generation models, and 3) mixing and coordinating multiple algorithms to complete the composition.[1]

Introduction

In recent years, more and more music creators are experimenting with AI for music composition [1]. Recent models such as MusicVAE [2] and the Seq-Attn model [3] mainly focus on improving the algorithmic performance of music generation. However, most music automation models still consider little about the actual needs of songwriters, which makes it challenging to make use of these AI models in practice. We consider human-computer co-creation systems as an effective solution. For example, the CoCoCo model [1][4] has an interactive Piano-Roll interface that makes the music generation algorithm Coconet [5] easier to use. Such systems clearly specify which parts of the composition should be done by humans and which parts could be automated, reducing the difficulty of controlling computational models via human-computer interfaces.

Psychological studies have long argued that natural language interpretation is a central organizing element of human learning and reasoning [6]. Thus, one possible solution for interacting with AI models is to use natural language through conversational systems. [7] To this end, we contribute the COSMIC conversational interface for human-AI music co-creation. The system undertakes a series of tasks, including melody composition, melody modification, lyric composition and lyric modification. In this paper, we focus on presenting its conversational approach to naturally control AI models for music composition. Figure 1 shows an example COSMIC screen.

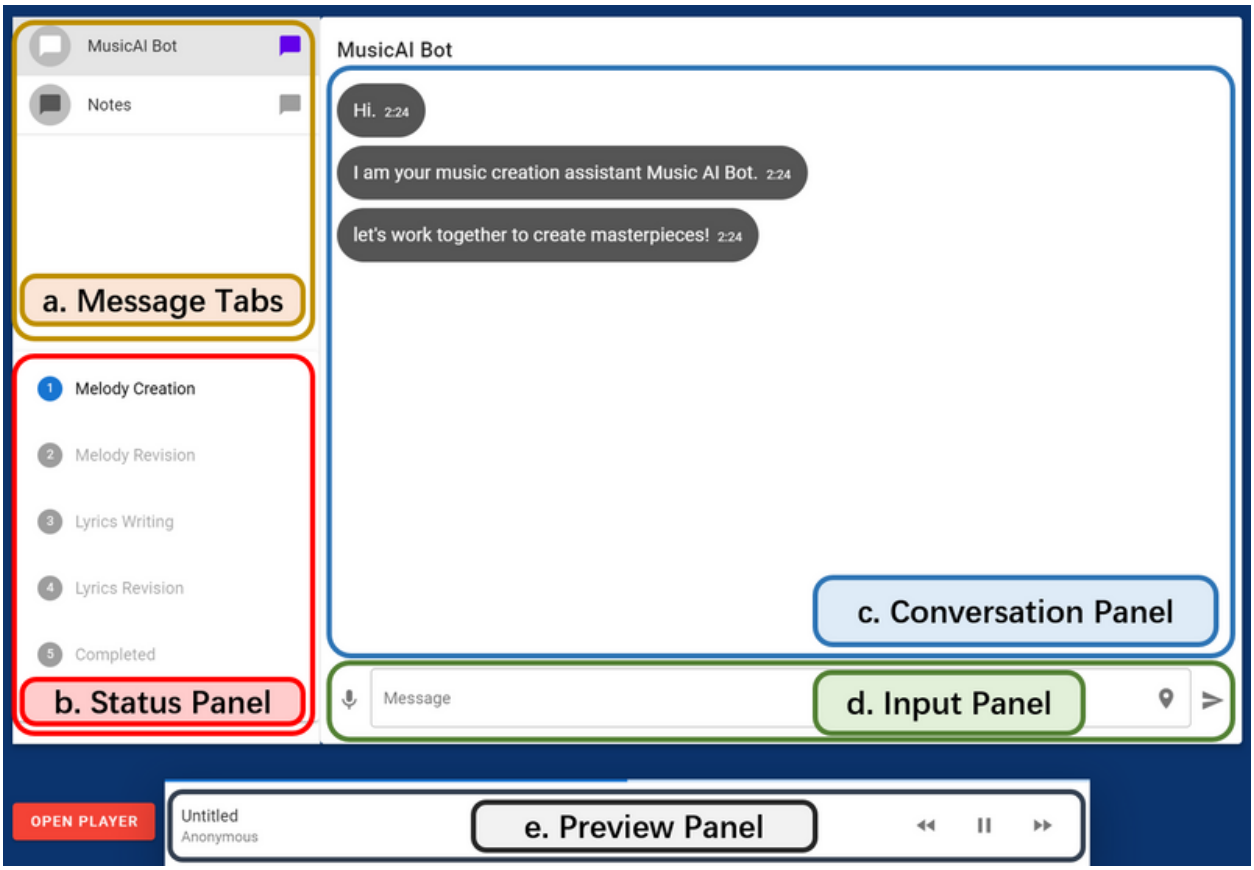


Figure 1
An example COSMIC screen.

Building a dialogue system and controlled music generation are the two main aspects of COSMIC's design. We used a dialogue tracker to make the dialogue system adaptable to the needs of music composition. Specifically, the dialogue system creates a form instance for each music creation. This form instance records the status of the music creation and parses the user's needs into form-filling actions, which will be elaborated upon in Section 3. Once this form is completed, COSMIC passes the information to the back-end model responsible for music generation and lyric generation. In the current version of COSMIC, we use BUTTER [8] as the model for music generation, which is a controlled music generation model capable of using natural language as a condition. We also train a controlled lyric generation model based on GPT-2, which will be elaborated in Section 2.4.

Architecture

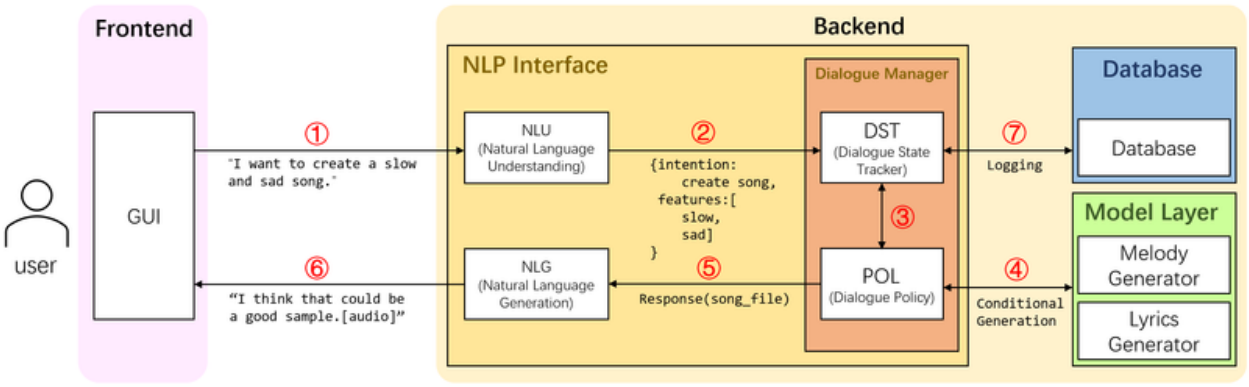


Figure 2
A diagram of COSMIC's handling of a single round of dialogue.

Natural Language Understanding (NLU)

COSMIC converts the process of music co-creation into the process of filling out a form, as shown in Table 1. The function of the NLU module is to convert the natural language into table entries of the creation form and give it to the Dialogue State Tracker (DST) module to be stored in the database. The NLU module parses two main attributes from the text input: *user purpose* and *keywords*. We compute the sentence representation using BERT [9] to decide which intention the user's purpose belongs to. We extract keywords using a POS algorithm. We consider the keyword extraction task as a POS task and use another BERT instance to complete it.

Table 1

Name	Content
Session ID	
Note Density	
Pitch Variance	
Rhythm Variance	
Genre of Lyrics	
Keywords of Lyrics	

An overview of the slot-filling form.

Dialogue State Tracker

The DST is designed to track the current dialogue state. It can be regarded as a finite-state machine. We design six different states for the DST module, as shown in Table 2.

Table 2

	State Name	Description
0	Start State	A new session is created and the metadata is initialized.
1	Generate Melody	COSMIC works with the user to determine the necessary conditions for melody generation, after which the melody is generated.
2	Revise Melody	COSMIC and the user determine the conditions that need to be modified, and the music is adjusted by modifying the high-level representation of the music.
3	Generate Lyrics	COSMIC determines the first line of the lyrics and keywords with the user, after which the lyrics are generated.
4	Revise Lyrics	COSMIC and the user decide on the lyrics that need to be changed, after which the changes are made.
-1	End State	The session is closed.

All states of the DST module.

We constrain the behavior of the DST such that the DST module is in one and only one state at each moment. In other words, the DST can be regarded as a finite-state machine with certain restrictions on state transitions.

Dialogue Policy

The dialogue policy module (POL) is mainly responsible for deciding the actions of COSMIC. We designed three different categories of response actions, which are: (1) Asking for more information; (2) Conditional generation and return of results; and (3) Returning a regular message.

The POL generation action is implemented through a simple rule-based method. When POL receives a command (e.g., “generate a melody"), it first checks the completeness of the slot-filling co-creation form. When the form is incomplete, POL asks the user for more information; when the form is complete, POL interacts with the model layer to get the data. Usually, POL also returns a short message to indicate that COSMIC has completed the corresponding processing.

Model Layer

The model layer of COSMIC is a separate module, which communicates with the POL module through a pre-agreed interface. From a software-engineering point of view, COSMIC allows all text-conditioned music generation models and text-conditioned lyric generation models to be connected to COSMIC in a plug-and-play manner, making COSMIC easily extensible. In the current version of COSMIC, we use two existing models to put into the model layer and retrain them to better fit our needs.

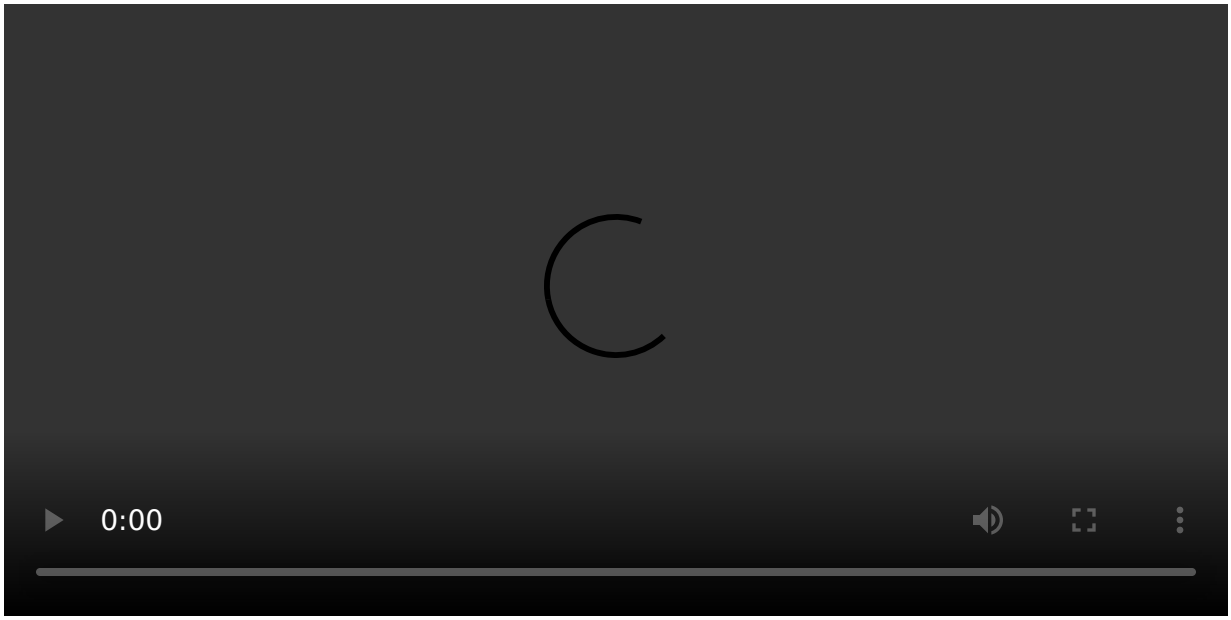
- **BUTTER** [8]. We use a variant model of BUTTER, BUTTER-Variance, as the back-end model for controllable music generation and music revision. The main difference between BUTTER-Variance and the original BUTTER is that BUTTER-Variance consists of two symmetric VAEs (instead of one VAE and two LSTMs), which are used to extract representations from music data and textual descriptions, respectively. A part of the latent vector computed by the two VAEs is subsequently aligned by a linear transformation to accomplish a weakly supervised representation decoupling. For the controlled music generation task, BUTTER-Variance performs a sampling from the already trained VAE and passes it to the decoder to obtain a piece of music; for the controlled music modification, BUTTER-Variance receives input from the text VAE and changes the latent vector of the music VAE by cross-modal alignment, thus modifying the original music representation. For a detailed description of BUTTER, please refer to the original paper [8].
- **CoCon** [10]. We use a variant of the GPT-2 based controlled text generation model CoCon, CoCon-Variance. The main difference between CoCon-Variance and CoCon is that CoCon-Variance achieves controlled generation of lyrics by constructing domain-

specific training samples (instead of long article sequences). CoCon-Variance accepts keywords and genres in natural language form as constraints, and COSMIC passes the user-defined guide text, keywords, and genre together into CoCon-Variance for training. For a detailed description of CoCon, please refer to the original paper [10].

Natural Language Generation

The NLG module is not used when the POL module only returns audio and when lyrics are returned. When POL needs to return a message, it passes the command to the NLG module for processing. In the current version of COSMIC, NLG is based on a corpus that holds a number of predefined samples. When POL asks NLG for output, NLG looks for sentences from the corpus that meet the conditions and outputs them after inserting and replacing keywords.

Demo



Video 1
A demo video of COSMIC.

Conclusion and Discussion

In this paper, we propose COSMIC, an NLP-mediated human-computer interaction-based music co-creation system. COSMIC enables users to perform controlled music co-creation with AI models through dialogue models. COSMIC still has some problems, such as the limited state space of the automaton-based dialogue state tracker, the weak practical performance of the music generation model, the simplicity of the music creation steps, and the performance of text responses generated by the NLU module.

In future research, we will improve the music co-creation process; we will also try training the dialogue model using an end-to-end approach to make the performance of the response text more natural.

Acknowledgements

Yixiao Zhang is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by the China Scholarship Council and Queen Mary University of London.

Footnotes

1. Code and demo are available at https://github.com/ldzhangyx/music-nlp-chatbot. ↩

Citations

1. Huang, C.-Z. A., Koops, H. V., Newton-Rex, E., Dinculescu, M., & Cai, C. J. (2020). AI Song Contest: Human-AI co-creation in songwriting. *ArXiv Preprint ArXiv:2010.05388*. ↩
2. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning* (pp. 4364–4373). PMLR. ↩
3. Jiang, J., Xia, G., & Berg-Kirkpatrick, T. (2020). Discovering music relations with sequential attention. In *Proceedings of the 1st workshop on nlp for music and audio (nlp4musa)* (pp. 1–5). ↩
4. Huang, C.-Z. A., Hawthorne, C., Roberts, A., Dinculescu, M., Wexler, J., Hong, L., & Howcroft, J. (2019). The Bach Doodle: Approachable music composition with machine learning at scale. *ArXiv Preprint ArXiv:1907.06637*. ↩
5. Huang, C.-Z. A., Cooijmans, T., Roberts, A., Courville, A., & Eck, D. (2019). Counterpoint by convolution. *ArXiv Preprint ArXiv:1903.07227*. ↩
6. Chin-Parker, S., & Cantelon, J. (2017). Contrastive constraints guide explanation-based category learning. *Cognitive Science*, *41*(6), 1645–1655. ↩
7. Smith, E. M., Williamson, M., Shuster, K., Weston, J., & Boureau, Y.-L. (2020). Can you put it all together: Evaluating conversational agents' ability to blend skills. *ArXiv Preprint ArXiv:2004.08449*. ↩

8. Zhang, Y., Wang, Z., Wang, D., & Xia, G. (2020). BUTTER: A representation learning framework for bi-directional music-sentence retrieval and generation. In *Proceedings of the 1st workshop on nlp for music and audio (nlp4musa)* (pp. 54–58). ↩

9. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *ArXiv Preprint ArXiv:1906.08237*. ↩

10. Chan, A., Ong, Y.-S., Pung, B., Zhang, A., & Fu, J. (2020). CoCon: A self-supervised approach for controlled text generation. *ArXiv Preprint ArXiv:2006.03535*. ↩