Computer Engineering Department

MusicGuru: Multi-task Multimodal Representation Learning for Music Applications

Project Advisor: Vijay Eranti

Basapur, Bhuvana Gopala Krishna (MS Software Engineering)
Gurumurthy,Arpitha (MS Software Engineering)
Lalwani, Mayuri (MS Software Engineering)
Mishra, Somya (MS Software Engineering)

Introduction

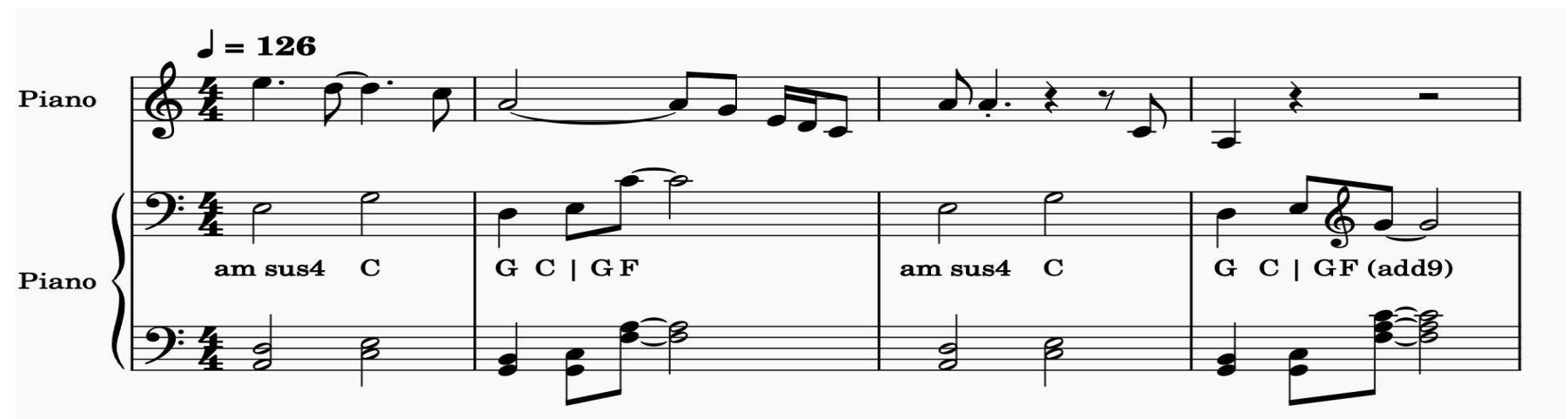
'MusicGuru' is an AI based application designed to cater music related tasks with Deep Learning techniques using State of The Art model. It's a one stop solution for music composers, DJs, music lovers to experiment and explore new melodies in minutes compared to days.

MusicGuru
Music starts here!

- Creativity of present musicians is limited to one or two genre/culture due to steep learning curve. MusicGuru will provide them a unique platform to expand their creativity to other genres by combining cross culture m
- MusicGuru has the ability to generate different tunes for the same melody every time the user clicks on it.
- User can share their composition with friends using Twitter.
- MusicGuru allows the user to save their composition to the local system or save it to the cloud.

Symbolic Music

- Symbolic music representation is the language of music. It involves encoding the components of music such as notes, pitches and rhythms in a human/machine readable format.
- MIDI (Musical Instrument Digital Interface) is one such format.
- Use MuseScore to view MIDI as sheet music.



MusicGuru Applications

- Melody Autocomplete: Users can input a few notes and MusicGuru will generate new melody on top of the input.
- Harmonization: MusicGuru can generate a new chord progression using the same melody.
- Melody generation: MusicGuru can generate a new melody for the existing chord progression.
- Song remixing: MusicGuru can change the Pitch of the input music while keeping the rhythm constant or change the Rhythm while keeping the Pitch constant.
- Melody Mixer: Using Magenta's MusicVue.js framework from Google, MusicGuru combines and transforms two different melodies by blending them at any percentage.

Architecture

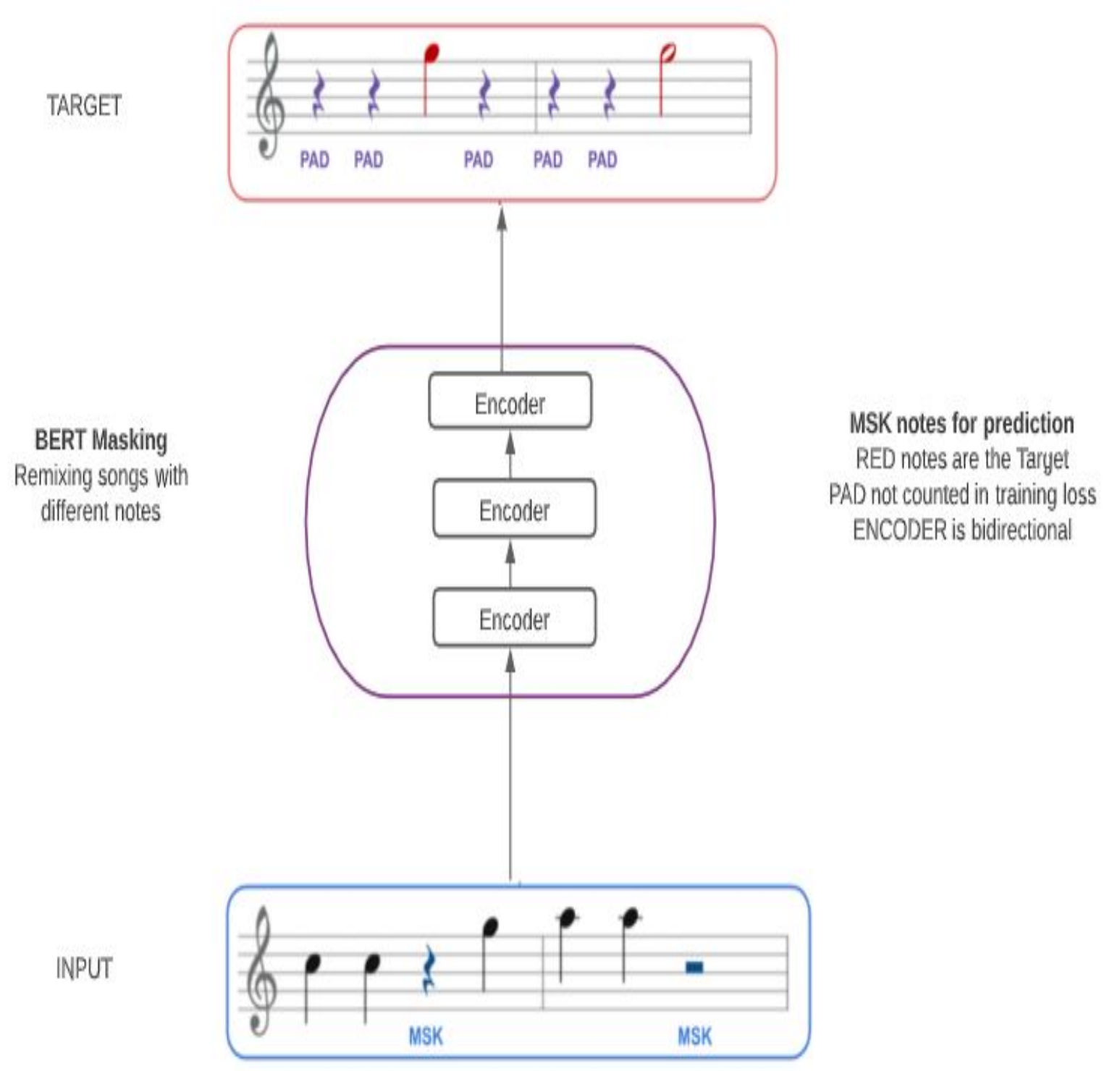
Transformer Variations

- Sequence to Sequence Translation (Seq2Seq): A class of neural networks that transforms a given input sequence into a different sequence. They particularly specialize in translation tasks in which a series of words in a language is translated into a series of words in a different language. It consists of an encoder and a decoder. The encoder accepts the input and converts it into a higher dimensional vector which is fed into the decoder to generate the output.
- Bert: BERT stands for Bidirectional Encoder Representations from Transformers. It is pre-trained on a large corpus of unlabelled text including the entire Wikipedia and Book Corpus. It specializes in predicting masked or missing tokens. Bi-directionality helps in providing complete context.
- TransformerXL: A class of neural networks that specializes in text generation. It consists of a forward directional decoder.

Mapping Applications to Models

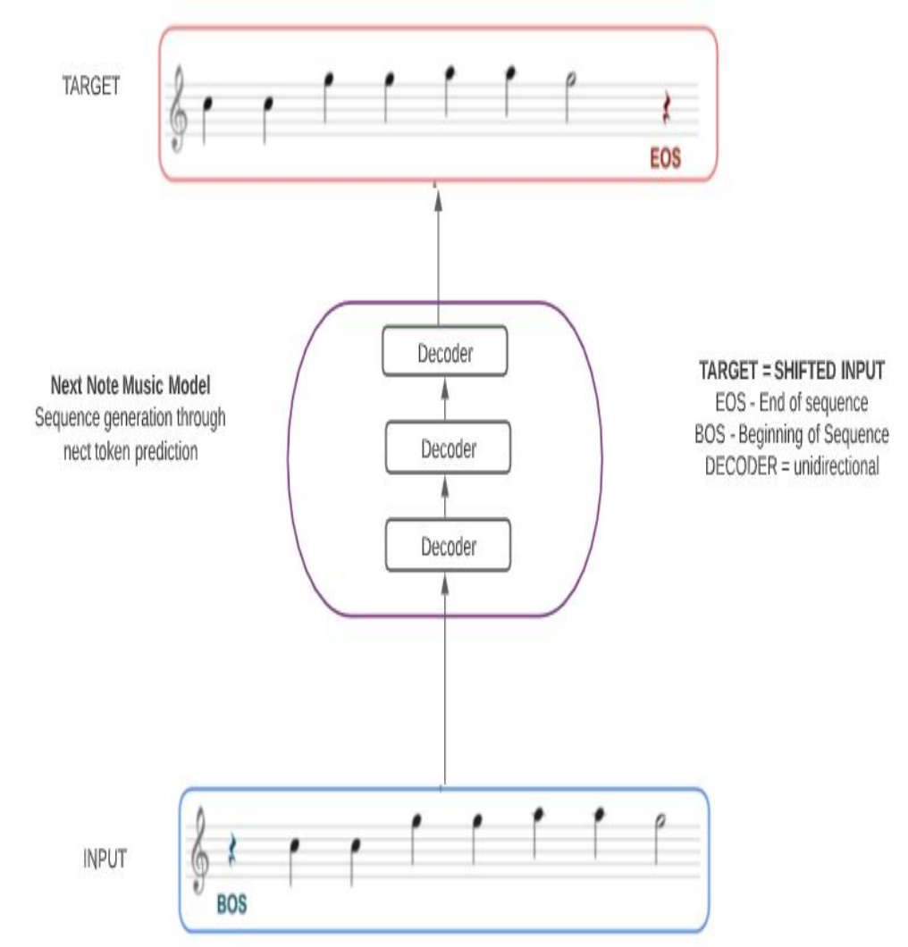
- Melody Autocomplete is implemented using TransformerXL which is great at next sequence generation.
- Song Remixing is implemented using BERT which is great at filling in the blanks. Certain parts of the song can be masked and use BERT to generate new variations.
- Harmonization and Melody Generation is implemented using Seq2Seq which is great for translation tasks. It is used to translate melody to chords and vice versa.

BERT



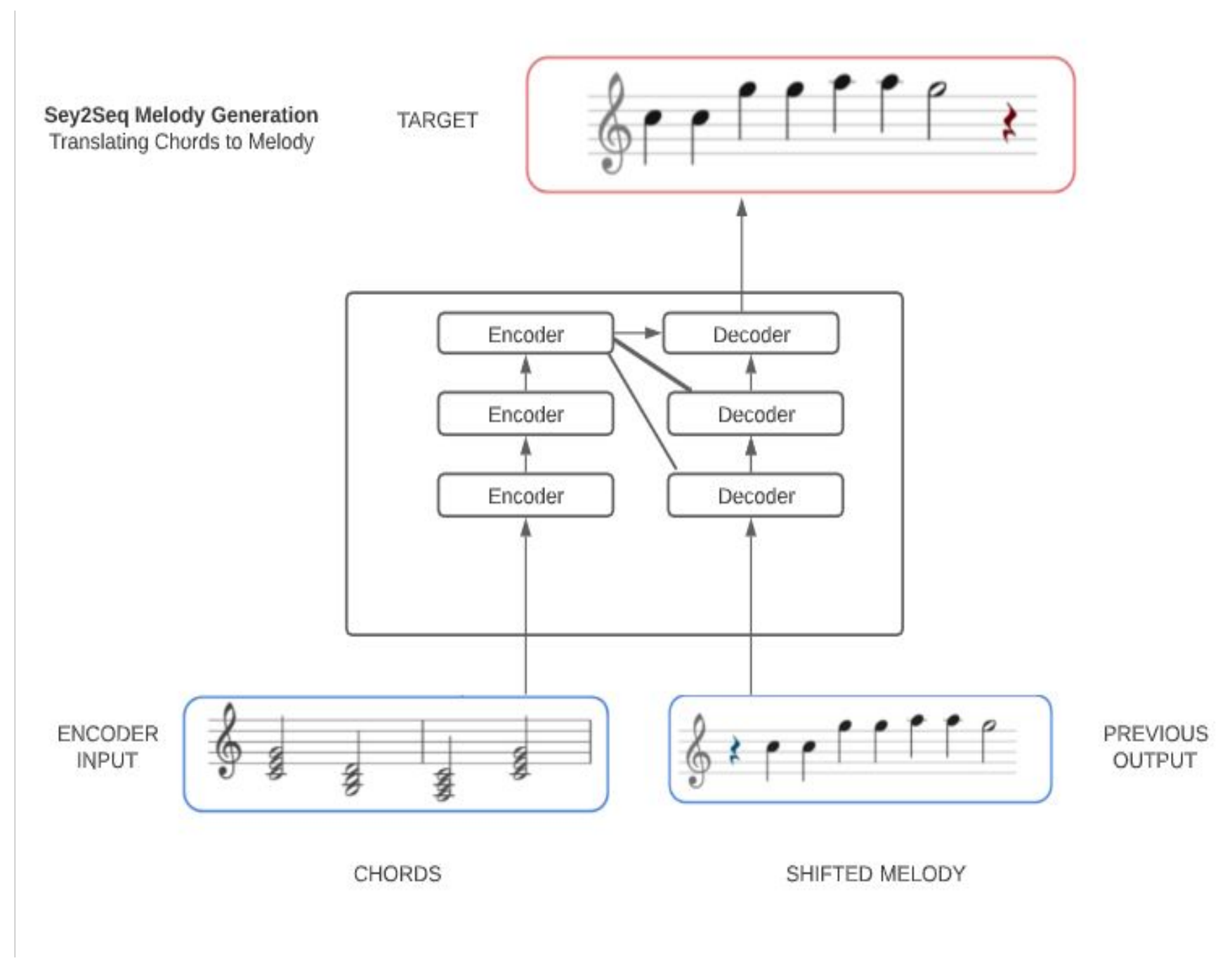
- We can mask some notes and BERT will fill in empty spaces.
- The Encoder is trained to predict the correct notes (red color notes in the output), whenever it sees a masked token (blue notes in the input)

Transformer XL



The Decoder is trained to predict the next token by shifting the target over by one.

Seq2Seq



- Very similar to language translation, Chords are translated to Melody or vice versa.
- The model consists of Encoder and Decoder blocks.
  - Encoder blocks are single bi-directional attention layers which can see all the input tokens.
  - Decoder blocks are double-stacked forward attention layers. These blocks use both the output from encoders and the previous output as context to generate melody.
  - The Decoder is unable to see the future tokens - they can only see the previous and current tokens.

Overall Architecture

- Bi-directional encoders are reused for training the BERT model.
- Similarly, the forward decoder layers can be reused for training the TransformerXL task.

If input is masked ('msk'), train the encoder.

If input is shifted ('lm'), train the decoder.

If input contains both translation input ('enc') and previous tokens ('dec'), train both encoder and decoder.

- MusicGuru has an authentication service using AuthO in order to respect user privacy.
- The model is deployed on the flask server and the predictions are stored in the S3 bucket with an idea to make retrieval faster and save space.
- MusicGuru is deployed on GCP for public access. Try it at http://34.72.49.148:8080
- Watch free demos of MusicGuru in action at http://34.72.49.148:8080/demo

Summary/Conclusions

The goal of this project is multimodal music understanding using intelligent DL techniques.To take it one step further we built a model that can jointly handle multiple music related downstream tasks to solve issues concerning extensive data and computational requirements..

MusicGuru leverages the powerful Transformer architecture and implements several music related tasks like Melody autocomplete, generate melodies and harmonization, Song Remixing.

Key References

[1]https://towardsdatascience.com/a-multitask-music-model-with-bert-transformer-xl-and-seq2seq-3d80bd2ea08e

[2]https://medium.com/@torinblankensmith/melody-mixer-using-deeplearn-js-to-mix-melodies-in-the-browser-8ad5b42b4d0b

[3] Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T. and Liu, T.Y., "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training", 2021, arXiv preprint arXiv:2106.05630.

[4] Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. 2020. "Pirhdy: Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic musi"c. In Proceedings of the 28th ACM International Confer- ence on Multimedia, pages 574–582.

Acknowledgements

We would like to thank our Professor Vijay Eranti for inspiring us and guiding us at every step in our journey. His invaluable support helped us in shaping our project.

We would like to also thank Professor Dan Harkey for his invaluable insights to plan and document all of our observations.