MusicGuru

Multi-task and Multimodal Representation Learning for Music

ADVISOR- *Prof. Vijay Eranti*

TEAM INVINCIBLES
Arpitha Gurumurthy
Bhuvana Basapur
Mayuri Lalwani
Somya Mishra

What is MusicGuru?

MusicGuru is an application that uses multitasking and multimodality for performing various music related tasks.

It uses a pre-trained transformer-based (Encoder Decoder) architecture.

The model is deployed on flask server which comes up on port 5000. The UI comes up on port 8080.

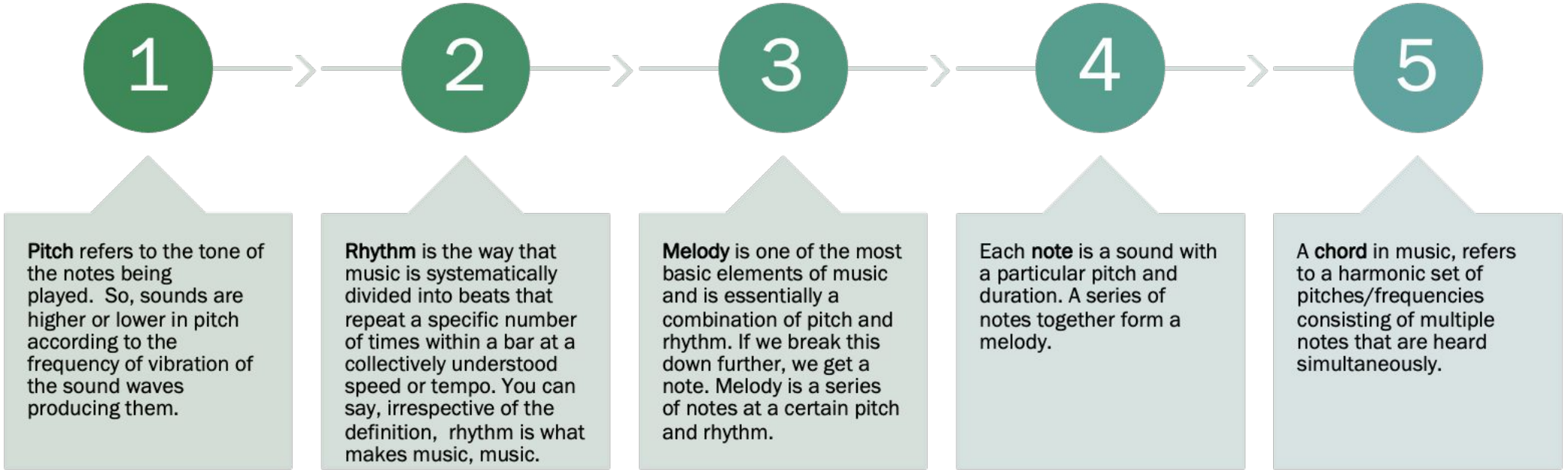
The predictions are then stored on the S3 bucket with an idea to make retrieval faster and save space.

MusicGuru also has an authentication service using AuthO in order to respect user privacy.

It has twitter integrated so the users can share their work.

Deployed on GCP at http://34.72.49.148:8080.

Music glossary



MELODY AUTOCOMPLETE

HARMONIZATION

MELODY GENERATION

SONG REMIXING

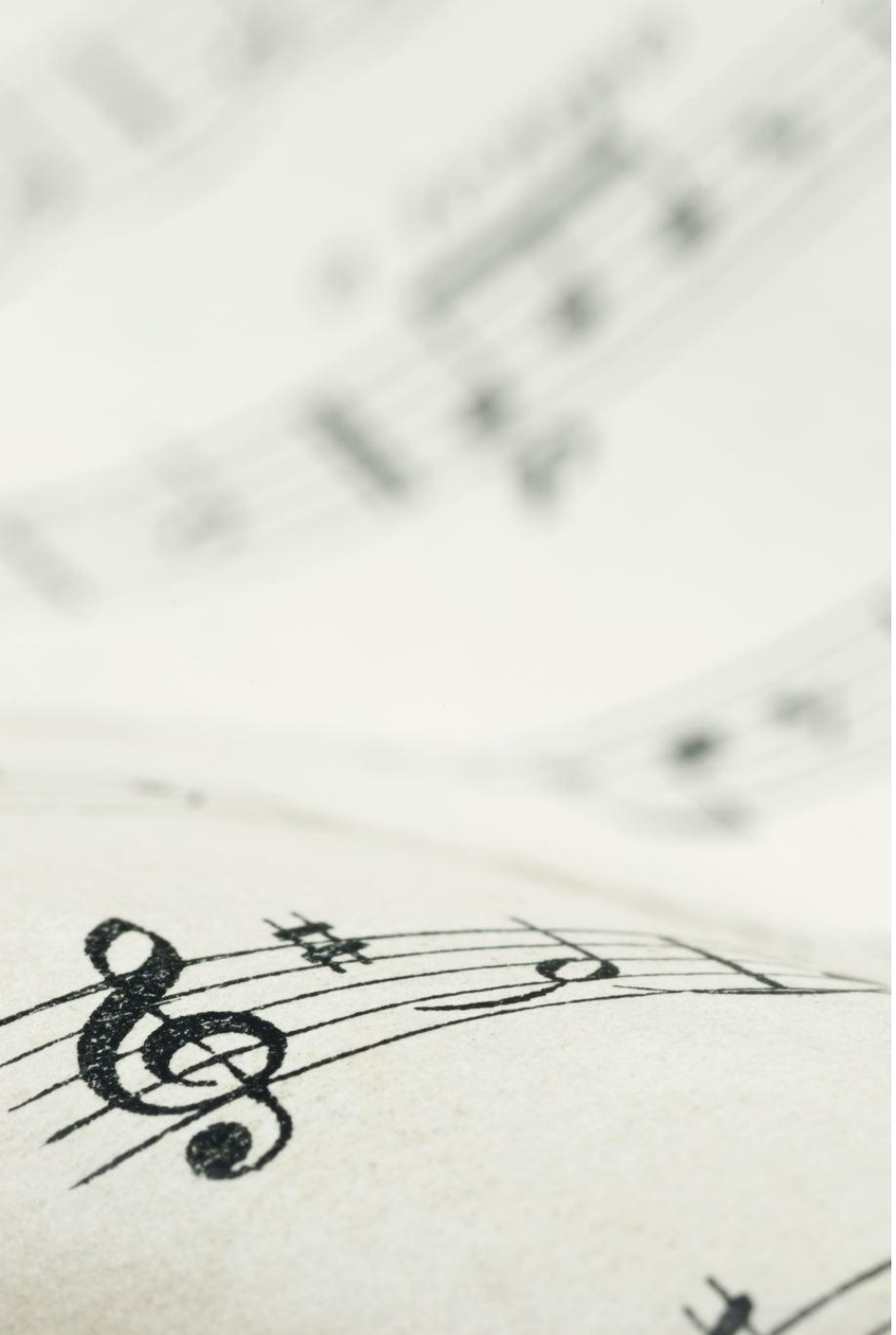
MELODY MIXING

What does MusicGuru offer?

MusicGuru's Applications

Melody Autocomplete

- Users can input a few notes of a tune to see how the application autocompletes the tune.
- They can do it as many times as they wish until they find ‘the one’ that resonates with them.
- The generated tune can be saved into their device or shared on Twitter directly from the app itself.



Generate Melody and Harmonize

- **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

Users can experiment with different pitches and rhythms for their tune.

They can even set the required pitch/rhythm they wish and experiment any number of times.

Melody Mixer

This task uses Magenta's MusicVAE.js, a web framework released by the Magenta research team at Google.

It lets us combine and transform two different melodies by blending them at any percentage

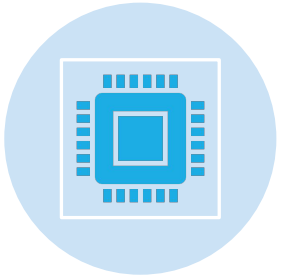
MusicVAE.js does this by running a deep neural network locally in our browser, using TensorFlow.js

Architecture

TransformerXL is great at sequence generation. This can be used to autocomplete a song idea → **Melody autocomplete**

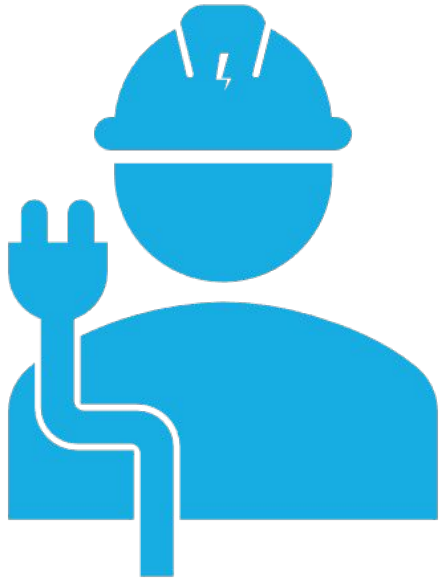
BERT is great for filling in the blanks. We can erase certain parts of the song and use BERT to generate a new variation → **Song remixing (pitch and rhythm)**

Seq2Seq is great for translation tasks. We can use this to translate melody to chords and otherwise → **Harmonization and Melody generation**



Our **Multitask model** is essentially the Seq2Seq architecture. The model is then modified and trained on different tasks.

Understanding Transformer XL



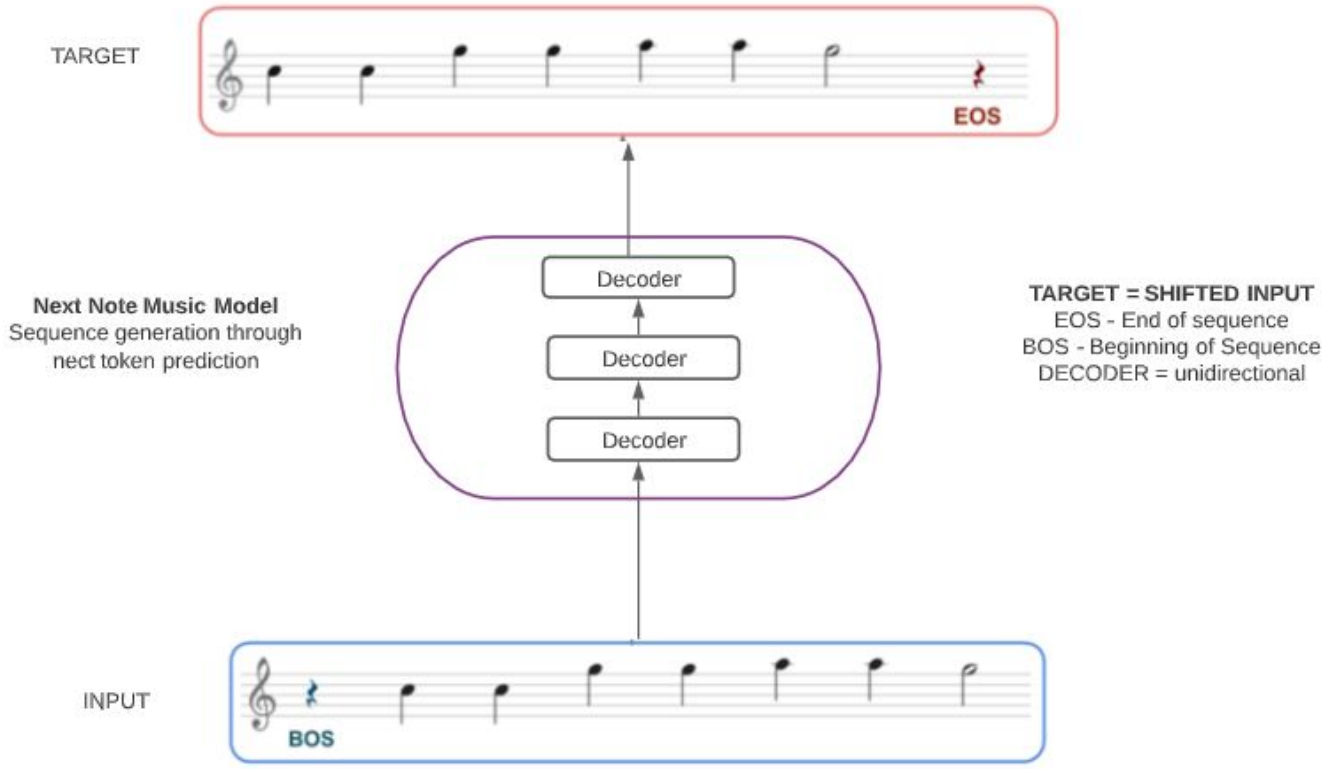
We know that Transformers help in capturing long-term dependency but is implemented with a fixed-length context (a long sequence is divided into fixed-length segments and each segment is processed separately). **They won't be able to model dependencies that are longer than a fixed length.**

TransformerXL was proposed to address these limitations:

1. Transformer-XL learns dependency that is about 80% longer than RNNs and 450% longer than vanilla Transformers
2. Transformer-XL is up to 1,800+ times faster than a vanilla Transformer during evaluation on language modeling tasks

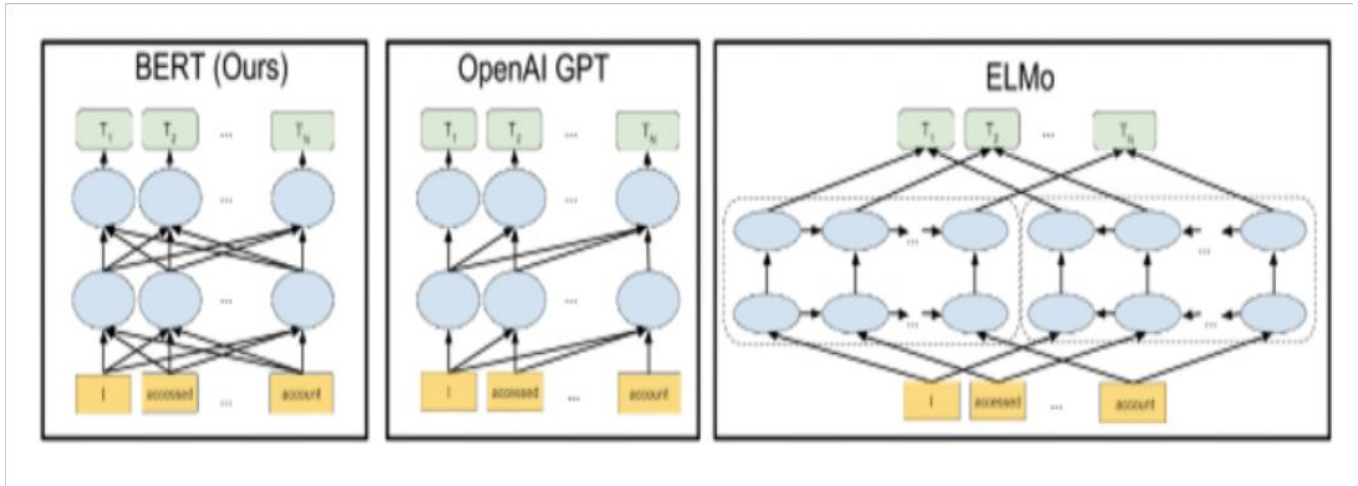
Transformer XL

THE DECODER IS TRAINED TO PREDICT THE NEXT TOKEN BY SHIFTING THE TARGET OVER BY ONE.



Understanding BERT

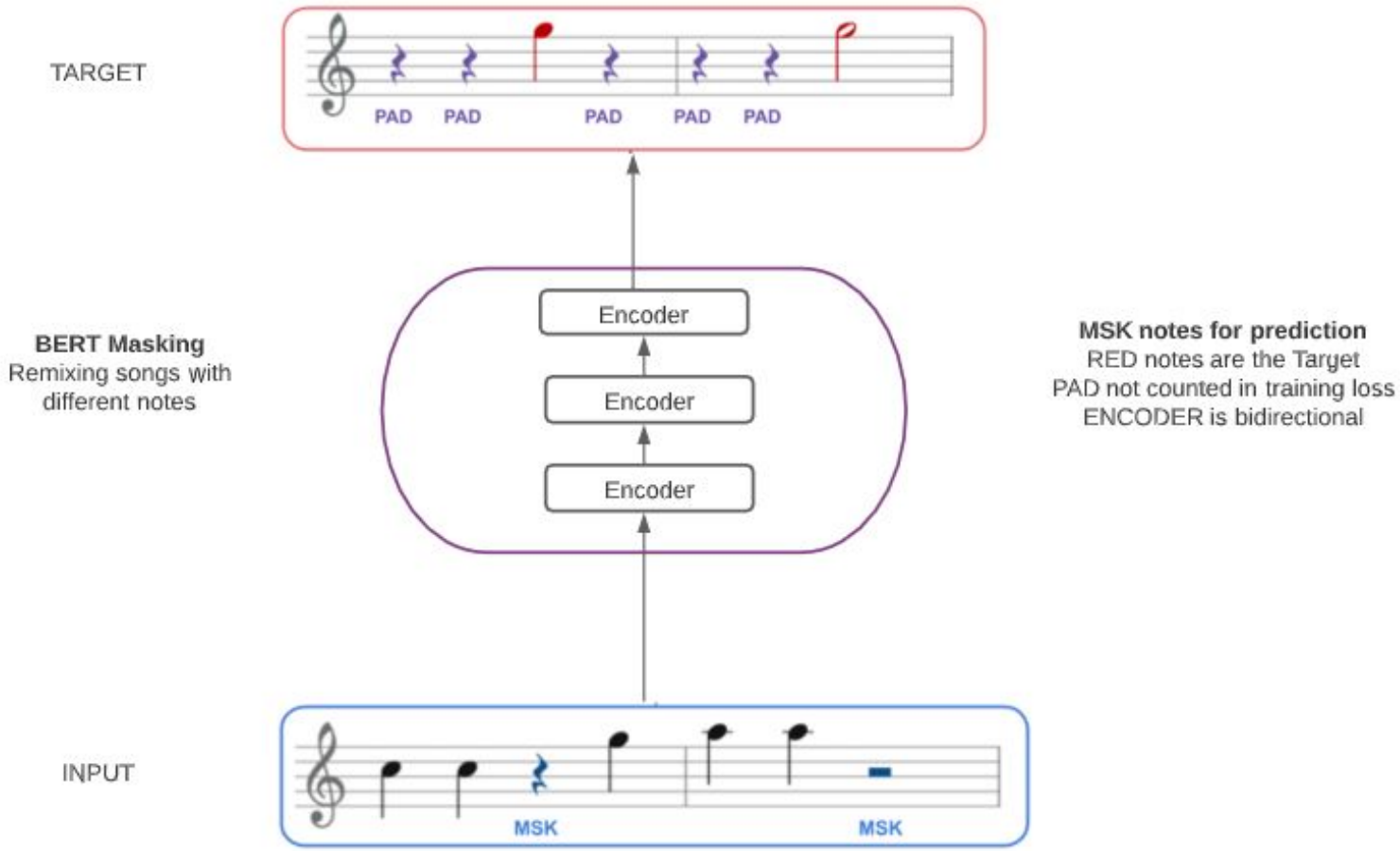
- One of the biggest challenges in natural language processing (NLP) is the shortage of training data.
- Bidirectional Encoder Representations from Transformers, or BERT was built to allow anyone to train a language model more efficiently since it provides contextual representations.
- It allows for contextual representations because it is deeply bidirectional and is trained on Wikipedia.

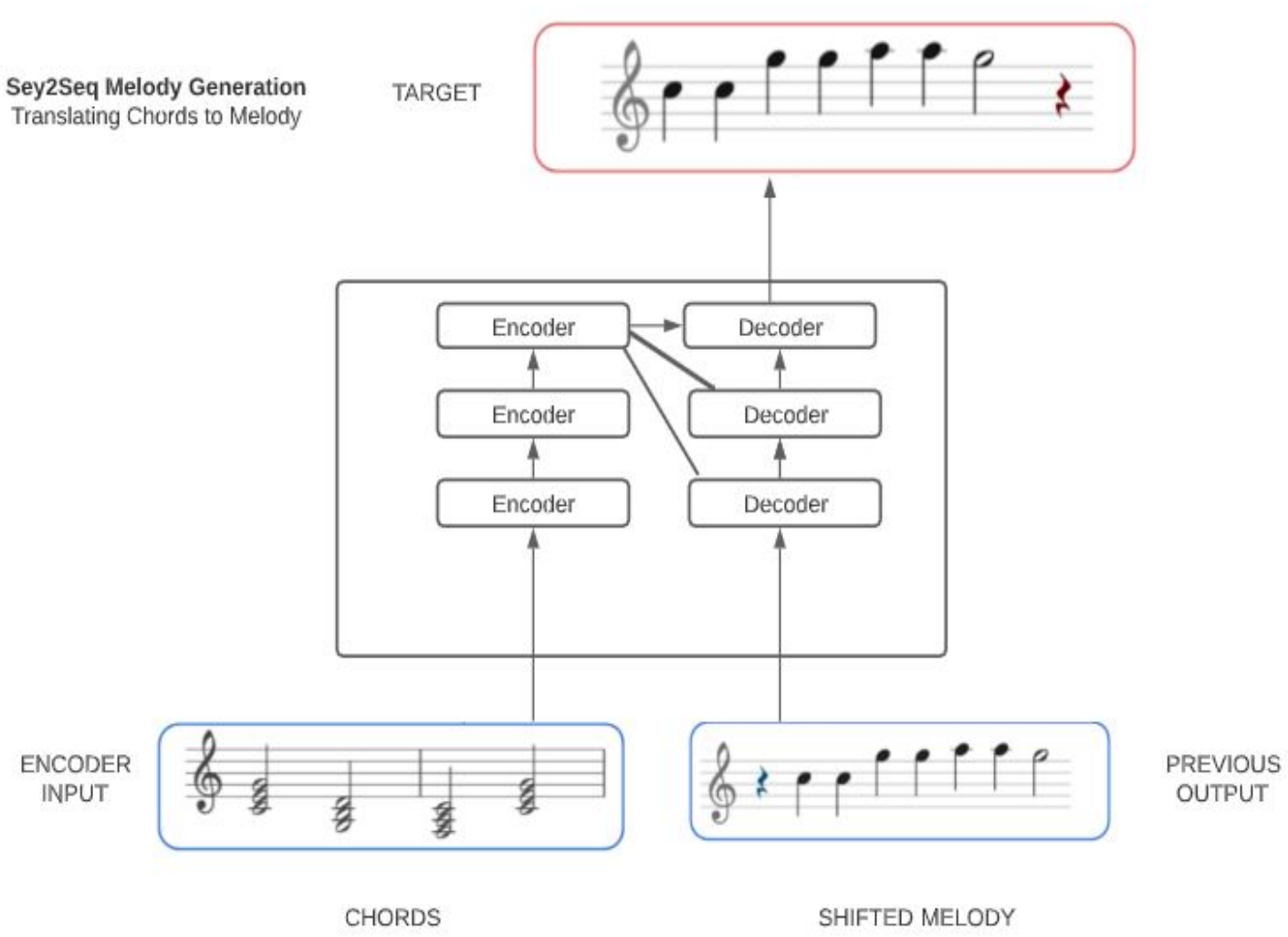


BERT

WE CAN MASK SOME NOTES AND BERT WILL FILL IN THE BLANKS

THE ENCODER IS TRAINED TO PREDICT THE CORRECT NOTES (RED), WHENEVER IT SEES A MASKED TOKEN (BLUE).





Seq2Seq

Very similar to language translation, chords are translated into melodies and vice versa.

Previous output is fed back into the decoder, so it gets more contextual information.

The model is split into encoder and encoder blocks.

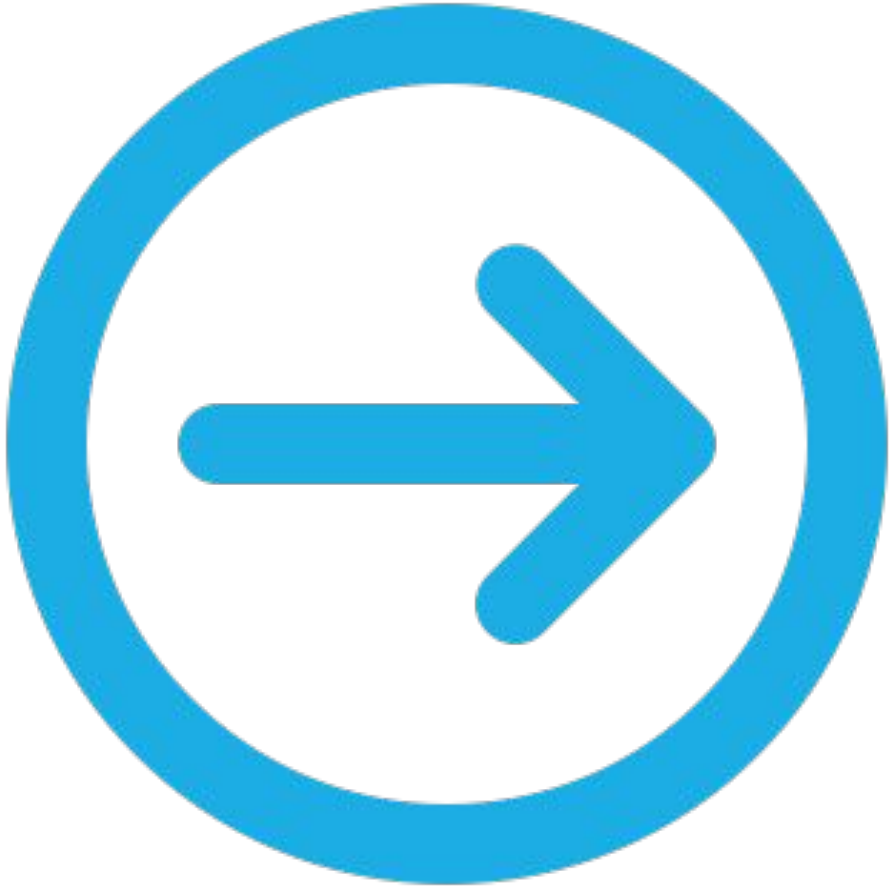
Encoder Blocks are single bi-directional attention layers. They can see all the input tokens.

Decoder Blocks are double-stacked forward attention layers. These blocks use both the encoder output and the previous output as context to predict the melody. The decoder is unable to see the future tokens, they can only see the previous and current tokens.

Seq2Seq - continued

Bi-directional encoders are reused for training the BERT model.

Similarly, the forward decoder layers can be reused for training the TransformerXL task.



Summary

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.

Try it out!

You can try out MusicGuru at http://34.72.49.148:8080!

Watch free demos of what MusicGuru does at http://34.72.49.148:8080/demo!

Thank You!

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.

References

- https://medium.com/@torinblankensmith/melody-mixer-using-deeplearn-js-to-mix-melodies-in-the-browser-8ad5b42b4d0b
- https://towardsdatascience.com/a-multitask-music-model-with-bert-transformer-xl-and-seq2seq-3d80bd2ea08e
- https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html
- https://ai.googleblog.com/2019/01/transformer-xl-unleashing-potential-of.html
- https://ai.googleblog.com/2017/04/introducing-tf-seq2seq-open-source.html