Title

Midigen: A Comprehensive Music Transcription Architecture

Who

- Luke Cheng, lcheng34
- Sanil Desai, smdesai
- Thomas Lin, tlin85
- Michael Lu, mlu85

Introduction

While there are models that convert raw audio to MIDI files or generate piano covers from audio files, there is no comprehensive pipeline that takes raw audio and directly transcribes music to arranged sheet music in a way that is true to the original source. We plan to make a model that takes in any raw audio file (any song or performance, live or otherwise) and converts it into transcribed sheet music. The preliminary data processing has been mostly implemented. There are papers that efficiently transcribe raw music files into midi format. We want to train a neural network that efficiently transcribes the midi format notes into a musical score divided by instruments/voice/etc. We will use a sequential multi headed neural net that passes the raw audio and transcribes that into a midi and then transcribes the midi into score. This will help reduce noise and ensure quality. The training data will be obtained from existing research papers and online databases such as Musescore and IMSLP.

Related Work

There are a few existing research papers that attempt to do similar work regarding music and its representations. For example, Pop2Piano uses a transformer architecture to generate piano covers of pop audio waveforms in MIDI format. Notably, this implementation circumvents melody or chord extraction modules, instead going directly to the arranged cover. Additionally, YourMT3+ employs a hierarchical attention transformer model to transcribe multi-instrument audio to MIDI. A unique feature of this implementation is the lack of voice separation preprocessing of the audio, as it can transcribe singing directly via cross-dataset stem augmentation.

- https://arxiv.org/pdf/2211.00895
- https://arxiv.org/pdf/2407.04822v3
- https://tanchihpin0517.github.io/PiCoGen/picogen2.html
- https://github.com/music-x-lab/POP909-Dataset

Data

We plan on employing publicly available datasets including MusicNet, Slakh, GuitarSet, EGMD, etc. as well as potentially scraping performances and transcriptions from MuseScore and IMSLP if necessary. These existing datasets are extensive (ex. Slakh has 2100 multi-track audio files, MusicNet has 330 pieces, EGMD has >45,000 files) and will not require significant preprocessing.

Methodology:

The architecture is an adaptation of the PiCoGen paper, which relies heavily on FluidSynth and SheetSage, which are older transcription models. We plan on using a perception encoder, which will use MERT-v1-95M (Music Understanding Model) with embeddings that are trained on understanding musical semantics. We will then use expressive performance diffusion, which will train a small diffusion model to predict velocity and micro-timing offset to account for human/live performance variability. This is to make the generated transcription sound more human. Then, we will use a MIDI-Conditional Latent Diffusion model, which will use our generated MIDI as a conditioning signal to account for details like pedalling and reverb. We will train the model by inputting the MERT embeddings, running raw audio through the MERT model to align the embeddings, then training the model to reconstruct piano MIDI from these embeddings. Then, we will use a dataset of piano performances aligned with audio and MIDI and train the diffusion model to account for offset and velocity deviations. Finally, for the actual renderer we will use a pre-trained model to render the final results (this is because training this will be unfeasible given time and compute constraints).

Metrics

Our model does not have the notion of accuracy because there’s a lot of freedom in how you can arrange a song and there isn’t some set of valid arrangements. There also is no objective criteria for what makes one arrangement better than another. Our goal is to make our model output piano arrangements that are both playable by a human and sound like our input song.

Our base goal for our model is to output sheet music with one note on each hand that resembles the input song, our target goal is to output a more complex version that resembles the song more, and our stretch goal will be to create a difficulty parameter that can be changed for how difficult you want the output to be. Because we are not transcribing, we won’t use a loss function because there is no ground truth to compare to. First, to ensure the output is playable, we will have a Playability Index, to penalize physically impossible configurations such as hand spans exceeding an octave or excessive note density. To quantify how good the music sounds we can use KL Divergence between our model’s output and professional sheet music and the entropy of our types of notes and combine these to form a metric. This will help our music sound less robotic. Finally, we want to allow creative freedom to some degree so we allow deviations but will penalize deviations that don’t match the pitch of the original song. This allows the model to generate deviations that penalizes any pitch deviations with the original song's key signature.

Ethics

1. What broader societal issues are relevant to your chosen problem space?

Music transcription directly relates to the societal issues of technology, authorship and creativity. By increasing automation in music - which is traditionally a very human-based field - this potentially displaces human involvement in music transcription and arrangement. While we intend for our tool to be more of an assistive technology rather than replacement for musical expertise, this would undoubtedly reshape human roles in the music industry, particularly in arrangement and transcription. We're also concerned about authorship and intellectual property. Specifically, there is clearly a risk of users uploading copyrighted material to bypass ownership boundaries and obtain complete musical scores. This also raises concerns about paid sheet music - if users can simply obtain a complete score from its audio, then why would users continue to pay to access scores?

2. What is your dataset? Are there any concerns about how it was collected, or labeled? Is it representative? What kinds of biases might it contain?

Our datasets - including MusicNet, Slakh, EGMD, GuitarSet, MuseScore and IMSLP - are likely skewed towards Western music. That is, they primarily represent Western genres such as classical, pop, jazz and typically use Western tonality and rhythmic structures (as well as notation). As such, the model may learn stylistic musical biases and therefore generalize poorly to music of other cultural systems, such as Jamaican reggae or Persian maqam.

Moreover, there are concerns about whether this data can be used for training purposes. While most datasets we are considering are public available, user-uploaded content from online platforms raise questions about whether the user had the correct rights/licenses to distribute the transcript. Even so, not all uploads may be legally or ethically sourced. Moreover, whether we are permitted to train on particular musical scores (e.g. if scores are only permitted for amateur musical use, and not commercial model training) may depend on the database and particular musical scores we access. Hence, we have to heavily consider the legality and ethics of each score.

Division of labor

Michael Lu: Model architecture, model tuning & poster graphics
Luke Cheng: Preliminary research, model tuning & poster graphics
Sanil Desai: Preliminary research, poster & model architecture
Thomas Lin: Final writeup, model tuning & poster content