MIDI Transformer Tokenization

Jonas Veit, NAMEN EINFÜGEN

January 8, 2025

Current workflow

1. Specify the MIDI data
2. Select a suitable Tokenizer. Right now we use REMI
3. train the tokenizer with BPE
4. save the tokenizer and vocabulary as json
5. generate training, validation and test sets via generate_train_val_test_sets.py we already augment data witihin this process if we want to.
6. Our GoePT expects integers as tokens and np.memmap expects binary files. We use np.to_file in order to create the right binaries

Different tokenizers

- For simple melodies: REMI or MelodyTokenizer.
- For more complex, polyphonic data: Consider PolyphonyTokenizer or NoteTokenizer.
- For structured MIDI data: Use Structured or TSD.
- For music generation: REMI or MIDI-Like might work well.

Things to keep in mind (Open Problems)

- We need to specify the sequence length
- We need to specify the embedding dimension
- We need to specfy the vocab_size
- When specifying a $seq_len > 32$ the data generation process fails. I have to investigate this. This problem is related to the function: miditok.split_files_for_training

- When we want to regenerate the train, val and test sets we need to manually delete the dataset_ folders! I will write some code for that soon.
- I think self.context_length from GoePT can be larger than the actualy seq_len specified within the tokenizer. What should we do here?