Maestro in TensorFlow: <https://www.tensorflow.org/tutorials/audio/music_generation>

Transformers for Music Generation: <https://magenta.tensorflow.org/music-transformer>

Embeddings:

* <https://github.com/midi-ld/midi2vec>
* <https://magenta.tensorflow.org/maestro-wave2midi2wave>
* <https://github.com/wazenmai/MIDI-BERT>
  + <https://towardsdatascience.com/byte-pair-encoding-the-dark-horse-of-modern-nlp-eb36c7df4f10>
* <https://towardsdatascience.com/practical-tips-for-training-a-music-model-755c62560ec2>
  + <https://github.com/bearpelican/musicautobot/blob/master/notebooks/data_encoding/Midi2Tensor.ipynb> (recommended polyphonic encoding)
  + <https://github.com/fosfrancesco/asap-dataset> metrically aligned note values
* <https://towardsdatascience.com/midi-music-data-extraction-using-music21-and-word2vec-on-kaggle-cb383261cd4e> (harmonic analysis and reduction)
* <https://towardsdatascience.com/representing-music-with-word2vec-c3c503176d52>
* Invariants:
  + look at music beat by beat
    - A note falls within a beat if start time < next beat and end time >= this beat (implicitly, end time > start time)
  + k longest note values matter (can be tuned; default is 4)
    - Time of a note is calculated by min(end time, next beat) - max(start time, this beat)
    - Sum time belonging to each pitch class (lose octave information)
* Use word2vec to embed?