Automatic Chord Arrangement from Melodies

Yulou Zhou, Shuxin Meng
{ylzhou, shuxin}@stanford.edu

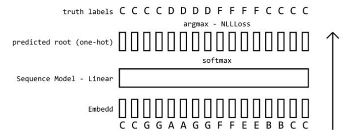
Introduction

- ❖ Problem: Music learners often encounter the challenge of arranging chords for melodies, because of their lack of music theory knowledge.
- ❖ An automatic chord arrangement tool that uses a LSTM model to generate chord for a given input melody.
- ❖ Chord arrangement involves both conventional rules and creativity.
- ❖ Ideal model: Generate chords that fit the melodies and follow the rules of progression, with a certain level of variation in outputs.
- ❖ Purposes:
  - ➢ Teach new learners of composition patterns of chord arrangement
  - ➢ Provide skilled composers with inspirations:

Models

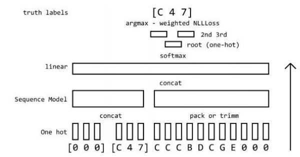
Baseline: sequence-to-sequence tagging

- ❖ Input: melody windows of a fixed size
- ❖ Numerical embedding
- ❖ All hidden states LSTM into a linear layer
- ❖ Output: one-hot matrix (input_size x 13)



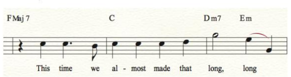
Final: sequence classification

- ❖ Input: previous chords, melody of current bar
- ❖ 2 LSTM for prev and melody respectively
- ❖ Concat the non-padded final state of LSTMs
- ❖ Predict the root note, then use the predicted root note to predict chord type



Data

- ❖ 2,252 lead sheets from Wikifonia (left image)
  - ➢ Posted by Music Audio Research Group in Seoul National University
- ❖ 1,802 sheets (72,418 bars) in training set, 450 (17,768 bars) in testing set
- ❖ Time signature, measure (bar), key, chord (root note, type) and note (root, octave, duration)
- ❖ Mostly one chord for each bar



1	4/4	[0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 10]	[12, 4, 7]
2	4/4	[9, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7]	[2, 3, 7]
3	4/4	[0, 0, 0, 0, 0, 0, 9, 9, 9, 9, 9, 10]	[10, 4, 7]
4	4/4	[9, 5, 5, 5, 5, 5, 5, 5, 5, 5, 7]	[5, 4, 7]
5	4/4	[9, 9, 9, 10, 10, 10, 9, 9, 9, 7, 7, 7]	[12, 4, 7]

Features

Convert all notes and rests to equal temperament numerical expressions (0-13, ignore octave)

Divided the duration of notes by each rhythmic unit (time signature), one note entrance per unit

Convert labelled chords to root notes and intervals (major→[+4, +7]) (right image)

Record previous chord progressions for each bar

Results and Discussion

- ❖ Baseline: Root note accuracy of 0.295
  - ➢ Result cannot be used practically, since does not specify initial position of chords
- ❖ Final: best result comes from Unidirectional LSTM with weighted loss

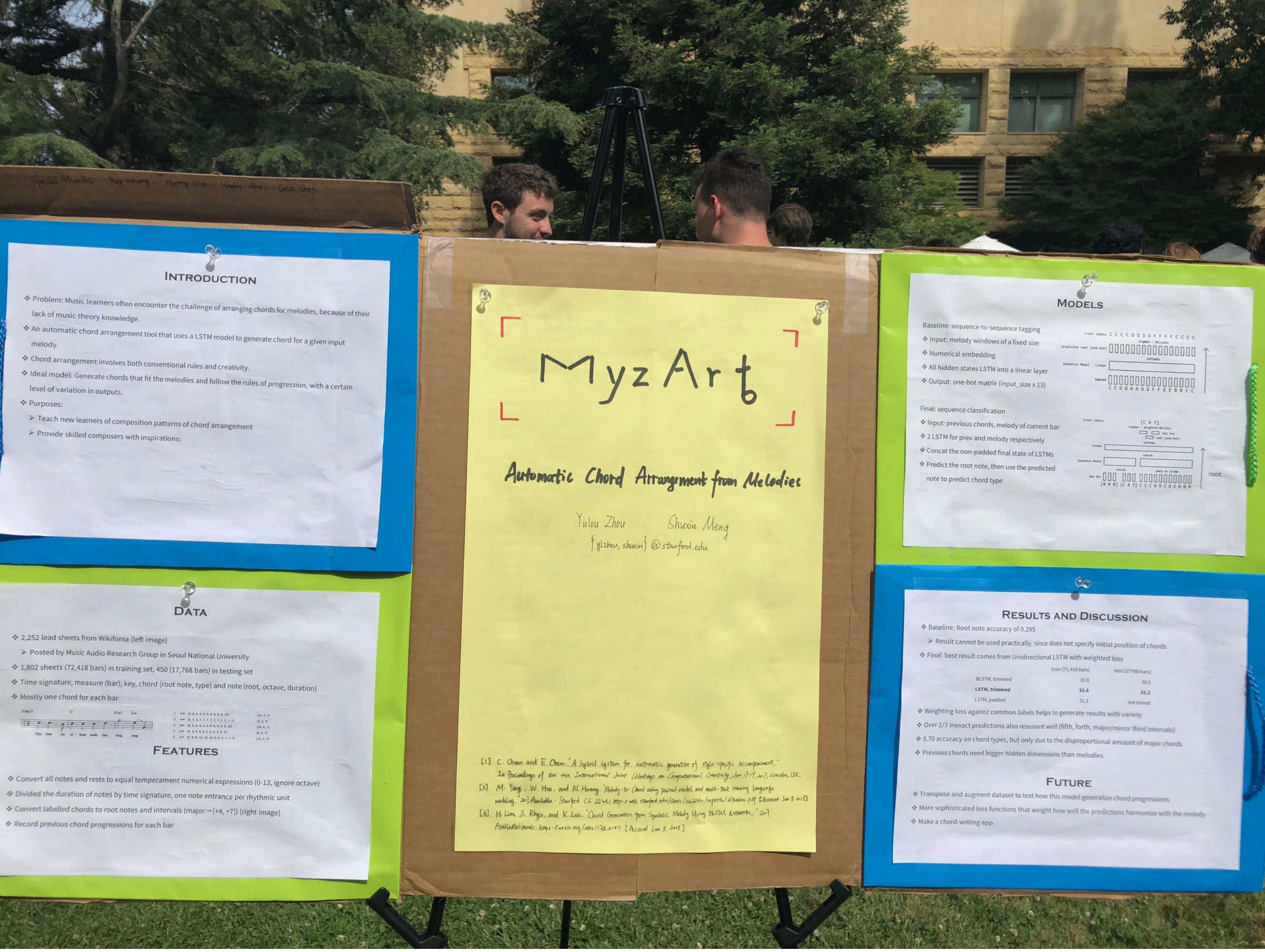
	train (72,418 bars)	test (17768 bars)
BLSTM, trimmed	33.0	30.5
LSTM, trimmed	**33.6**	**31.2**
LSTM, padded	31.1	not tested

- ❖ Weighting loss against common labels helps to generate results with variety
- ❖ Over 2/3 inexact predictions also resonant well (fifth, forth, major/minor third intervals)
- ❖ 0.70 accuracy on chord types, but only due to the disproportional amount of major chords
- ❖ Previous chords need bigger hidden dimensions than melodies

Future

- ❖ Transpose and augment dataset to test how this model generalize chord progressions
- ❖ More sophisticated loss functions that weight how well the predictions harmonize with the melody
- ❖ Make a chord-writing app

[1] Chuan, Ching-Hua, and Elaine Chew. "A hybrid system for automatic generation of style-specific accompaniment," In Proceedings of *the 4th International Joint Workshop on Computational Creativity*, Jun 17-19, 2017, London, UK.
[2] Yang, Mu-Heng, Wei-Ting Hsu, and Nicholas Huang. "Melody-to-Chord using paired model and multi-task learning language modeling," 2017. Available: Stanford CS 224N, http://web.stanford.edu/class/cs224n/reports/2742800.pdf [Accessed Jun 8, 2018]
[3] H. Lim, S. Rhyu, and K. Lee. "Chord Generation from Symbolic Melody Using BLSTM Networks," 2017. Available: arxiv, https://arxiv.org/abs/1712.01011 [Accessed Jun 8, 2018]



Introduction
Problem: Music learners often encounter the challenge of arranging chords for melodies, because of their lack of music theory knowledge.
An automatic chord arrangement tool that uses a LSTM model to generate chord for a given input melody.
Chord arrangement involves both conventional rules and creativity.
Ideal model: Generate chords that fit the melodies and follow the rules of progression, with a certain level of variation in outputs.
Purposes:
Teach new learners of composition patterns of chord arrangement
Provide skilled composers with inspirations:
Data
2,252 lead sheets from Wikifonia (left image)
Posted by Music Audio Research Group in Seoul National University
1,802 sheets (72,418 bars) in training set, 450 (17,768 bars) in testing set
Time signature, measure (bar), key, chord (root note, type) and note (root, octave, duration)
Mostly one chord for each bar
Features
Convert all notes and rests to equal temperament numerical expressions (0-13, ignore octave)
Divided the duration of notes by time signature, one note entrance per rhythmic unit
Convert labelled chords to root notes and intervals (major→[+4, +7]) (right image)
Record previous chord progressions for each bar
MyzArt
Automatic Chord Arrangement from Melodies
Yulou Zhou Shuxin Meng
{ylzhou, shuxin} @stanford.edu
Models
Baseline: sequence-to-sequence tagging
Input: melody windows of a fixed size
Numerical embedding
All hidden states LSTM into a linear layer
Output: one-hot matrix (input_size x 13)
Final: sequence classification
Input: previous chords, melody of current bar
2 LSTM for prev and melody respectively
Concat the non-padded final state of LSTMs
Predict the root note, then use the predicted note to predict chord type
Results and Discussion
Baseline: Root note accuracy of 0.295
Result cannot be used practically, since does not specify initial position of chords
Final: best result comes from Unidirectional LSTM with weighted loss
train (72,418 bars) test (17768 bars)
BLSTM, trimmed 33.0 30.5
LSTM, trimmed 33.6 32.2
LSTM, padded 31.1 not tested
Weighting loss against common labels helps to generate results with variety
Over 2/3 inexact predictions also resonant well (fifth, forth, major/minor third intervals)
0.70 accuracy on chord types, but only due to the disproportional amount of major chords
Previous chords need bigger hidden dimensions than melodies
Future
Transpose and augment dataset to test how this model generalize chord progressions
More sophisticated loss functions that weight how well the predictions harmonize with the melody
Make a chord-writing app