Computational Harmony

How to approach a complex musicological problem with machine learning

Gianluca MICCHI

Computer Music, Cambridge, 11/11/21

Outline

MUSICOLOGICAL PART

- What is harmony?
- Harmonic analysis and Roman numerals notation
- A musicologist’s algorithm

TECHNICAL PART

- Feature extraction
- Sequence learning
- Classification

CONCLUSIONS

- Statistical analysis of the results
- Looking at some examples
- roman.algomus.fr
- A glance at harmonisation possibilities

Music prerequisites

NOTES

- Pitch: the frequency at which the sound vibrates. Pitch distance is measured in semitones; twelve semitones make an octave, after which pitch names are repeated.
- Rhythm: succession of note onsets and durations, expressed in arbitrary units of measure (quarter note)

HARMONY

- Several notes played simultaneously (or almost) form a chord
- Chord quality: major, minor, diminished, seventh…
- Quality is determined by the relative distance between pitches. Eg, minor chord: [0, 3, 7] ↔ (root, minor third, fifth)

GOOD TO HAVE

- Music sheet reading

E -

root quality

Introduction

MELODY HARMONY

Take The “A” Train: $A\flat$, $B\flat^{7}$, $B\flat^{-7}$, $E\flat^{7}$

https://youtu.be/cb2w2m1JmCY

Introduction

same distance between roots, same function

I Got Rhythm: B♭, C-^{7}, F^{7}

Take The "A" Train: A♭, B♭^{7}, B♭-^{7}, E♭^{7}

https://youtu.be/oQdeTbUDCiw

Chord Symbols to Roman Numerals

Roman numerals help to bring forward the function of a chord in a piece
The key of a piece can't be determined by a single chord out of context.
Here, we assume that the key is known through other means

Chord symbol	e
Key	e
Roman Numeral	e: i

Chord Symbols to Roman Numerals

Roman numerals help to bring forward the function of a chord in a piece
The key of a piece can't be determined by a single chord out of context.
Here, we assume that the key is known through other means

Chord symbol	e/G
Key	e
Roman Numeral	e: i^6

Chord Symbols to Roman Numerals

Roman numerals help to bring forward the function of a chord in a piece
The key of a piece can't be determined by a single chord out of context.
Here, we assume that the key is known through other means

Chord symbol	G^7
Key	e
Roman Numeral	e: III^7

Chord Symbols to Roman Numerals

Roman numerals help to bring forward the function of a chord in a piece
The key of a piece can't be determined by a single chord out of context.
Here, we assume that the key is known through other means

Chord symbol	G^7
Key	C
Roman Numeral	C: V^7

I Got Rhythm:
B♭, $C\text{-}^7$, F^7 → I, ii^7, V^7 in B♭

Take The "A" Train:
A♭, $B\flat^7$, $B\flat\text{-}^7$, $E\flat^7$ → I, II^7, ii^7, V^7 in A♭

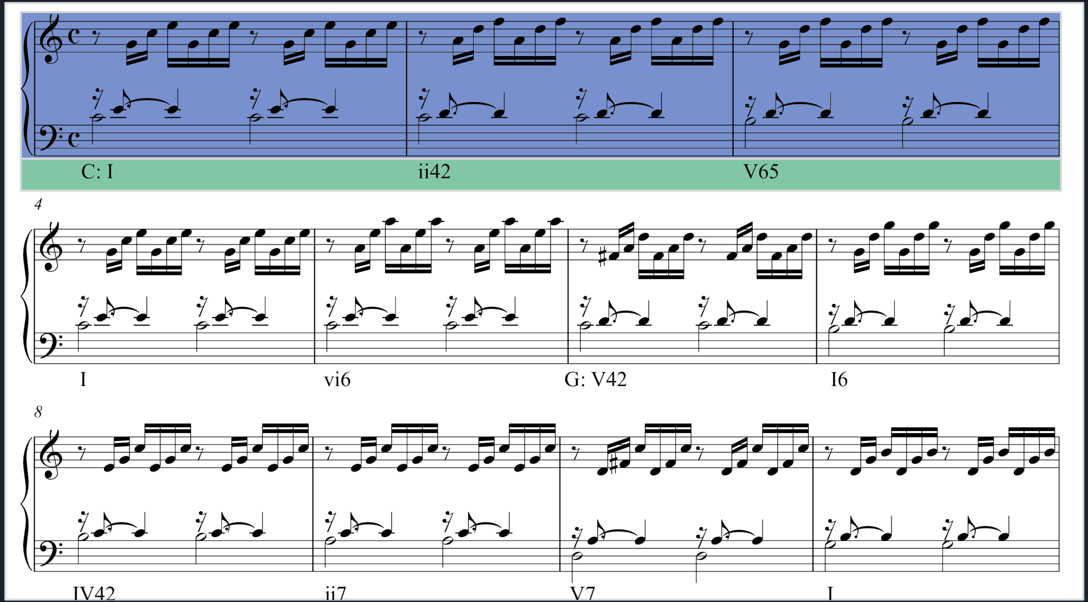
A Musicologist's Algorithm

1. Identify a Key
2. Segment Chords
3. Assign Chord Labels
4. Study the Progression

https://youtu.be/0egJr6nvCQI

Bach WTC I Prelude #01 in C

Input
Output



A Musicologist’s Algorithm - 2

NON-HARMONIC TONES

4. Identify a Key
1. Segment Chords
- Assign Chord Labels
3. Study the Progression ?

https://youtu.be/YmhvPzTZbAM

SCHUBERT Winterreise D911: No.12, Einsamkeit

A Computational Musicologist’s Algorithm

Rule-based algorithms look very hard to implement in this case, therefore we try a different approach: machine learning. Let’s take a step back and state the problem from an abstract perspective.

- the input data is a (long) sequence of tokens (valid for both audio and symbolic)
- the output data is also a long sequence of tokens
- alignment between input and output is of the utmost importance
- at every step of the sequence, the algorithm must select the correct output token out of a dictionary of available tokens → classification problem

Harmonic Analysis is a complex classification problem with sequences both in input and output

There is no known rule-based solution

ML architecture

1. Feature extraction
2. Sequence learning
3. Classification

Feature extraction

Examples of feature extraction:

- piano-roll notations (symbolic)
- spectrograms (audio)
- MFCCs (audio)
- CNNs (learned or not)
- autoencoder encoding
- ...

The goal of feature extraction is to take the information contained in the data point and to transform it into a sequence of features that can be interpreted by the rest of the machine learning algorithm.

It is sometimes hard to draw a line where feature extraction finishes and the sequence analysis starts.

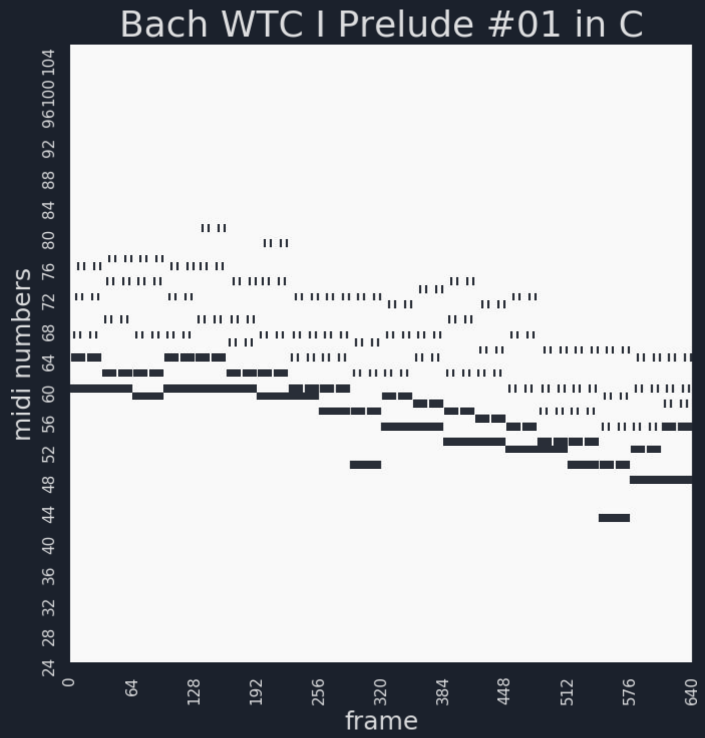
Feature extraction — input representation

Input: pianoroll (symbolic music)

- quantization on time axis (1/32 note)
- splitting pieces in shorter chunks
- multi-hot encoding

Issues

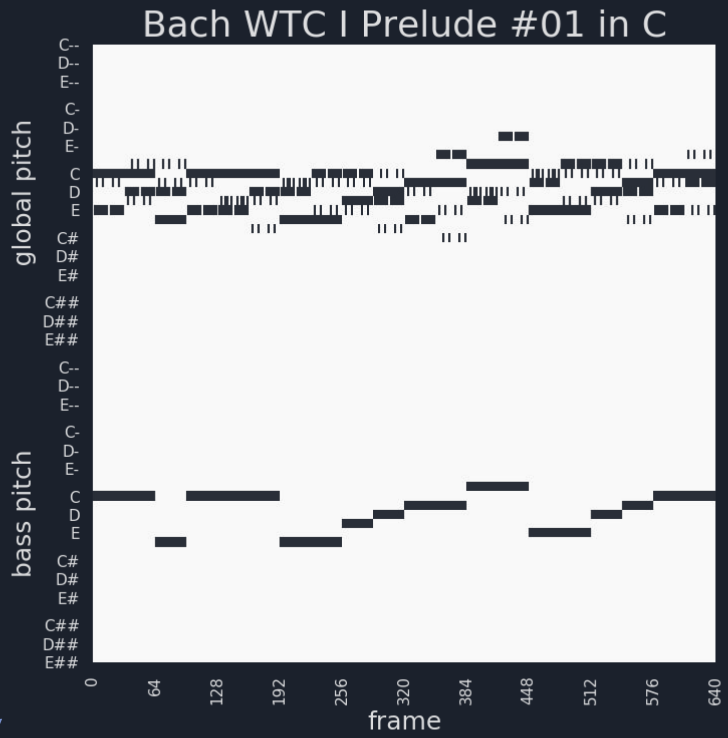
- non-invariant for octave transposition
- no pitch spelling: A♯ = B ♭
- notes held or repeated?



Feature extraction — input representation

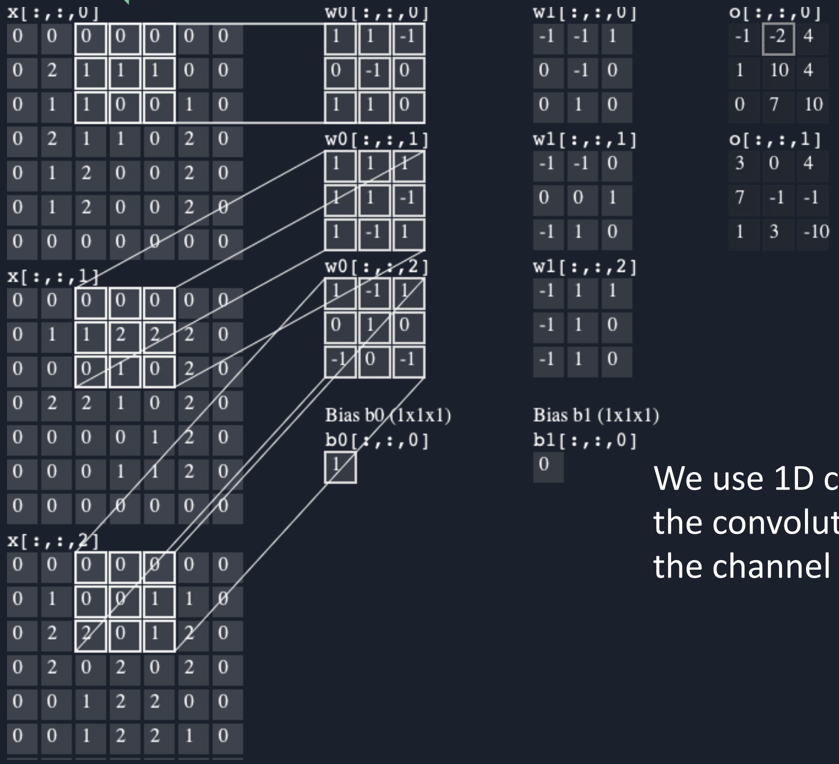
Improving on the pianoroll notation

	NO SPELLING	YES SPELLING
OCTAVE	Chromatic pitch, full 12 x 7 = 84 (MIDI numbers)	Pitch spelling, full 35 x 7 = 245 (too large a set?)
BASS	Chromatic pitch, bass 12 x 2 = 24	**Pitch spelling, bass** **35 x 2 = 70**
NEITHER	Chromatic pitch, class 12 x 1 = 12 (not enough info?)	Pitch spelling, class 35 x 1 = 35 (not enough info?)



See also https://transactions.ismir.net/articles/10.5334/tismir.45/

Feature extraction — CNN

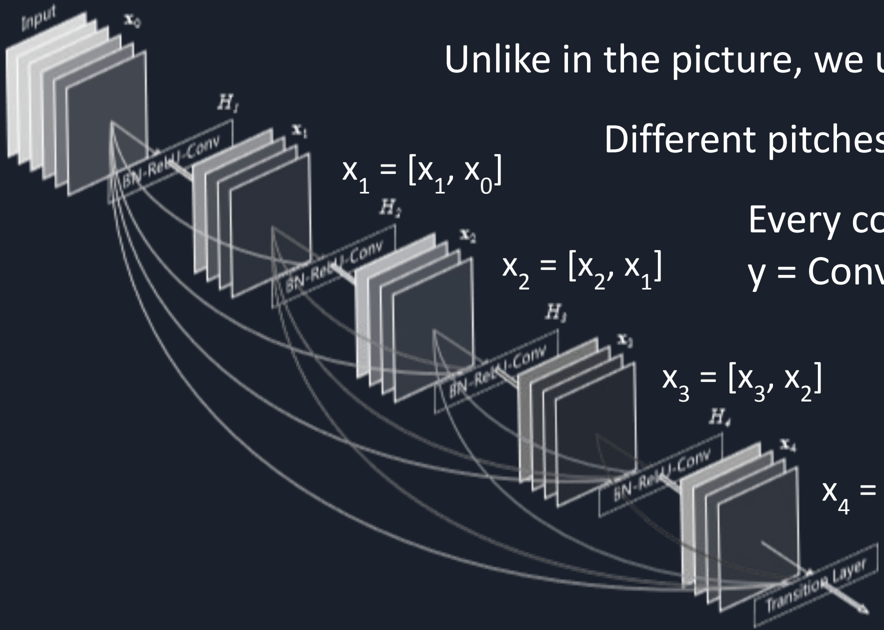


- Weight sharing: fewer weights to train
- Better properties wrt translation
- Possibility to stack many levels -> more abstract functions
- strong assumption that information has a spatial locality and coherence

We use 1D convolutions:
the convolution dimension is time,
the channel dimension is pitches.

Feature extraction — DenseNet

Improving on the basic CNN architecture



$x_1 = [x_1, x_0]$

$x_2 = [x_2, x_1]$

$x_3 = [x_3, x_2]$

$x_4 = [x_4, x_3]$

Unlike in the picture, we use 1D convolutions (time axis)

Different pitches go to different channels

Every conv is actually made of a two parts:

$y = \text{Conv1D}(1)(x_i)$; $x_{i+1} = \text{Conv1D}(k)(y)$

The network keeps all relevant information at all times

Three such blocks, separated by pooling layers

Final global context: 2 quarter notes

Image taken from https://arxiv.org/abs/1608.06993

Sequence learning

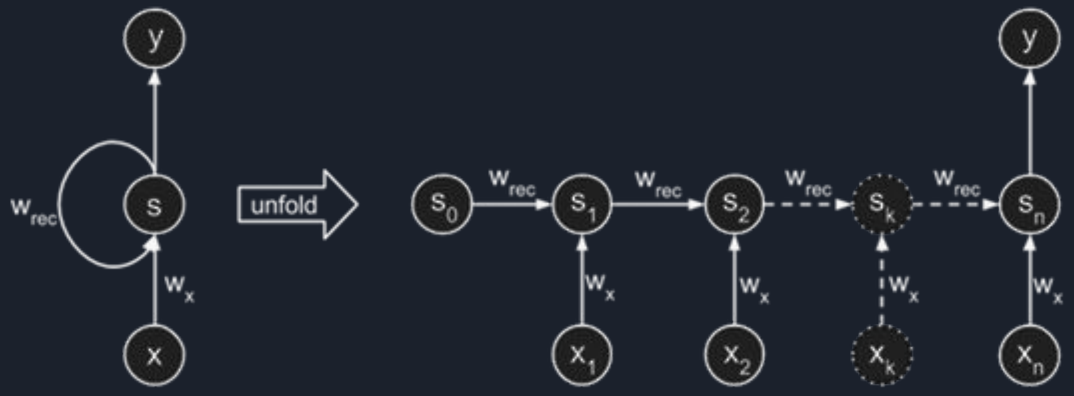
Examples of sequence learning algorithms:

- LSTM
- GRU
- Seq2Seq
- Transformer
- ...

Given a sequence of input features, analyse them to understand the mutual interaction at different times.

Especially in music, sequence learning is very important because it encodes all long-term information such as key analysis, harmonic progressions, etc.

Sequence Learning — RNN

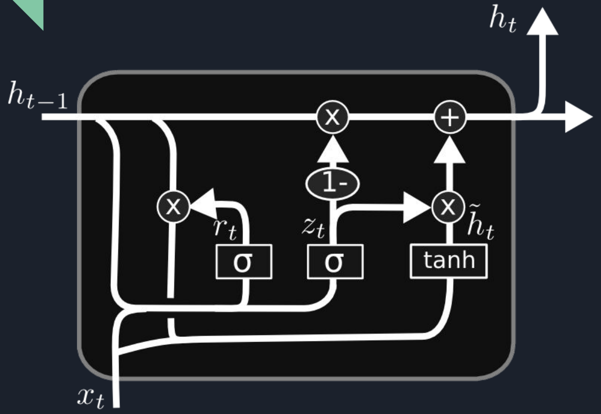


- Again weight sharing, albeit different
- Can keep information for many time steps
- Strong assumption that information has a temporal coherence

- Sequential, so comparatively slow to train
- This simple version is not used because of exploding and vanishing gradients over very long sequences

Sequence Learning — GRU

Improving on the basic RNN architecture



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$ *update*

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$ *reset*

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$ *candidate*

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$ *hidden / output*

- the hidden state h_t flows freely through the network -> good back-propagation
- with respect to LSTMs, GRUs have fewer weights (3 matrices instead of 4)
- we use a bidirectional layer, as we have knowledge of entire piece when analysing

see also https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Classification

The learned representation at every time step must be used to determine the chord label among the available ones.

This is typically done using a softmax over all available classes, but there are problems in this case due to the sheer size of the chord vocabulary.

Classification — Output representation

Let's do some math…

types of chord	12 x
inversions per chord	4 x
keys (C ♭ to C♯ and relatives)	30 x
scale degrees (1-7; ↓, o, ↑)	21 x
degrees for tonicised chords	21 =
	635,040

sneak peek into the data:
we have ca. 100k annotations

The number of output classes is too big!

1. Separate key from the rest:
   one classification with 30 classes
   one classification with ~20k classes
   https://transactions.ismir.net/articles/10.5334/tismir.65/
2. Make 5 separate predictions, one per each category
   https://transactions.ismir.net/articles/10.5334/tismir.45/
3. Make separate but coherent predictions
   https://archives.ismir.net/ismir2021/paper/000055.pdf

Meta-corpus

as published in May 2020, now ~40% larger

Dataset	Composer/s	Movements or equivalent	Quarter length	Measures	RNs
TAVERN	Mozart	10 theme and variations sets	7 712	2 773	8 779
	Beethoven	17 theme and variations sets	12 840	5 128	15 959
ABC	Beethoven	16 string quartets, 70 movements	48 811	15 881	29 652
BPS-FH	Beethoven	32 piano sonata first movements	30 992	9 420	11 337
Roman Text	Bach	24 preludes	3 168	819	2 165
	Various (19th C.)	48 romantic songs	8 326	2 791	5 283
Totals		201 scores	111 859	36 812	73 175

Devaney, J. et al., **Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis.** *(ISMIR 2015)*
Neuwirth, M. et al., **The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets.** *(FiDH)*
Chen, T.-P. and Su, L., **Functional Harmony Recognition of Symbolic Music Data with Multi-Task Recurrent Neural Networks.** *(ISMIR 2018)*
Tymoczko, D., et al., **The roman text format: a flexible and standard method for representing roman numeral analyses.** *(ISMIR 2019)*

Data is further augmented by transposing the pieces

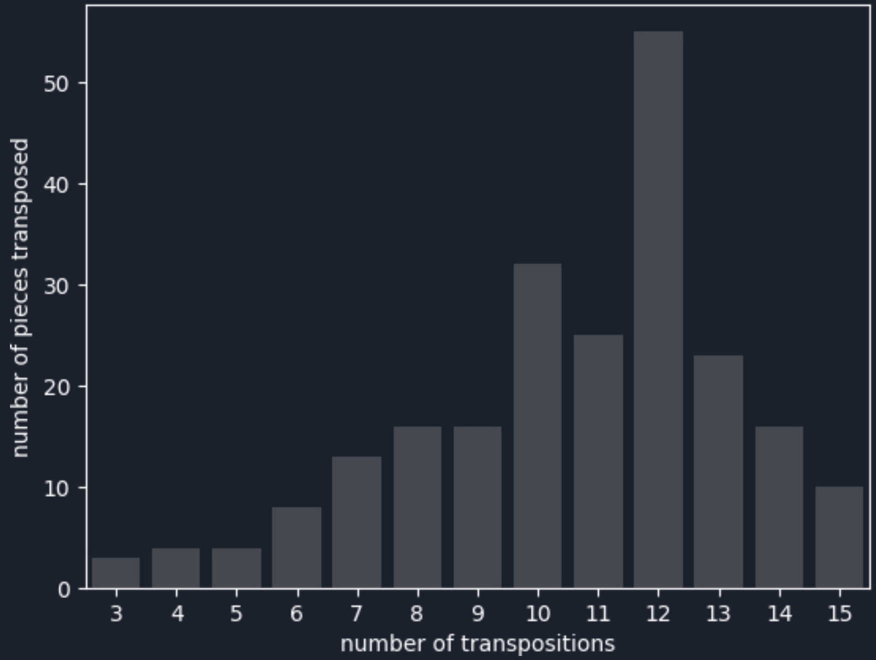
Data augmentation

Transposing pieces to different keys.

- all notes must have max ♭♭ or ♯♯
- keys must be inside the range

  C♭ to C♯ (and relatives)

Harmonically constrained pieces are more represented than daring ones!



Example of data

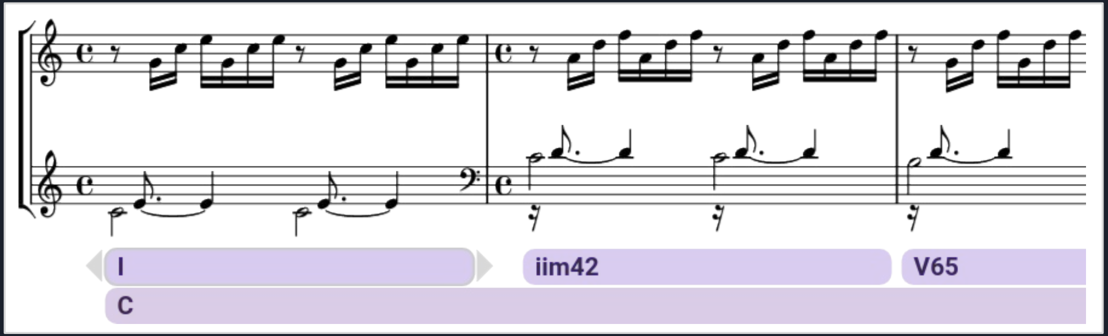
Roman Numeral annotations are given in three formats:

- rntxt, for human analysis

m1 b1 C: I	0.0,4.0,C,1,M,0
m2 b1 ii42	4.0,8.0,C,2,m7,3
m3 b1 V65	8.0,12.0,C,5,D7,1
m4 b1 I	12.0,16.0,C,1,M,0
m5 b1 vi6	16.0,20.0,C,6,m,1
m6 b1 G: V42	20.0,24.0,G,5,D7,3
m7 b1 I6	24.0,28.0,G,1,M,1
m8 b1 IV42	28.0,32.0,G,4,M7,3
m9 b1 ii7	32.0,36.0,G,2,m7,0
m10 b1 V7	36.0,40.0,G,5,D7,0
m11 b1 I	40.0,44.0,G,1,M,0
…	…

- csv, for training the model

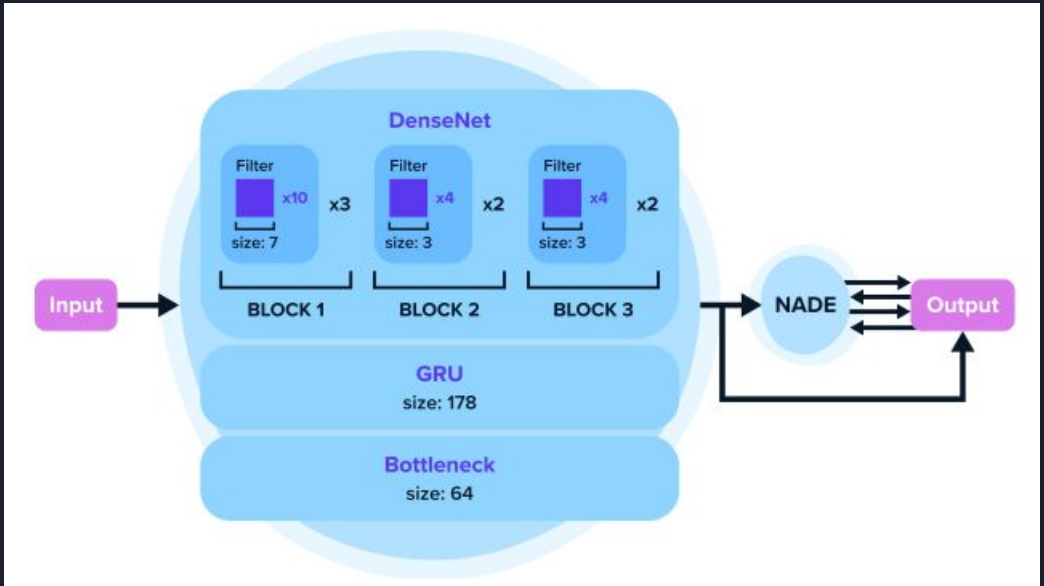
- json, for visualizing with Dezrann



Giraud, M. et al., **Dezrann, a Web Framework to Share Music Analysis.** *(TENOR 2018)*

scores in musicXML, not shown

Classification — The NADE



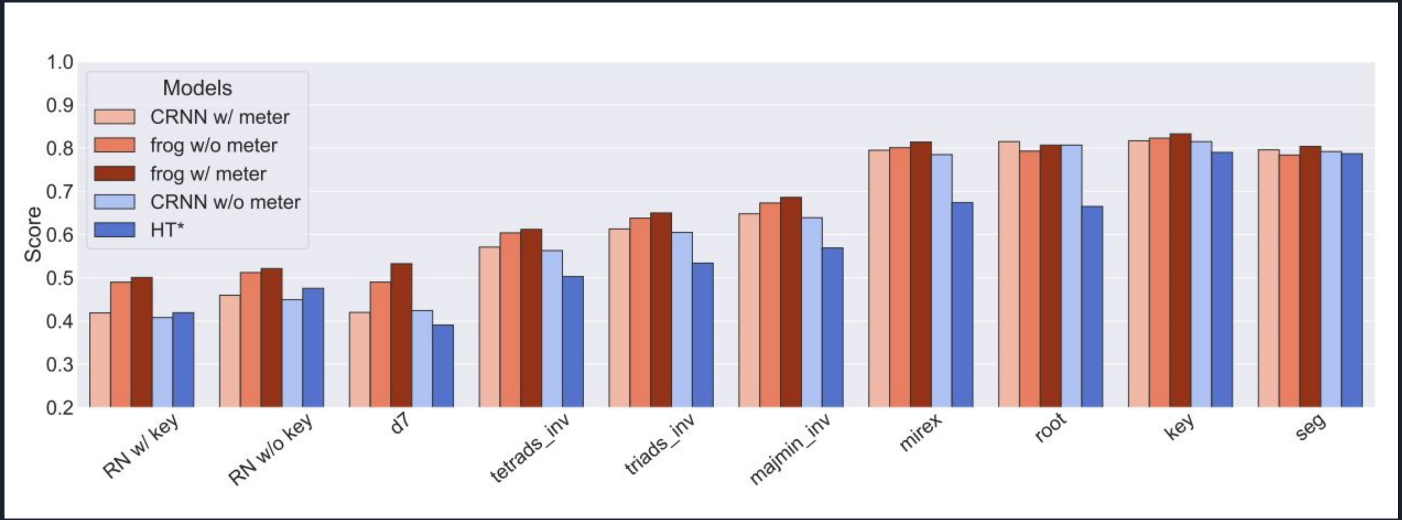
The NADE enforces the coherence between the labels; it has one hidden and one visible layer, connected by weights V and W and with biases given by the CRNN.

$$p(x_d|\boldsymbol{x}_{<d}) = \text{sigmoid}(\boldsymbol{V}_d \cdot \boldsymbol{h}_d + b_d), \qquad \boldsymbol{b} = \text{sigmoid}(\boldsymbol{\theta}_v \cdot \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\beta}_v),$$

$$\boldsymbol{h}_d = \text{sigmoid}(\boldsymbol{W}_{<d} \cdot \boldsymbol{x}_{<d} + \boldsymbol{c}), \qquad \boldsymbol{c} = \text{sigmoid}(\boldsymbol{\theta}_h \cdot \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\beta}_h).$$

See also https://archives.ismir.net/ismir2021/paper/000055.pdf

Conclusions — Accuracy of the results



See also https://archives.ismir.net/ismir2021/paper/000055.pdf

Conclusions — Analysis of the results

Root Coherence

The root is predicted directly but can also be derived using the other features. If the predicted root differs from the derived root then the labels are not coherent.

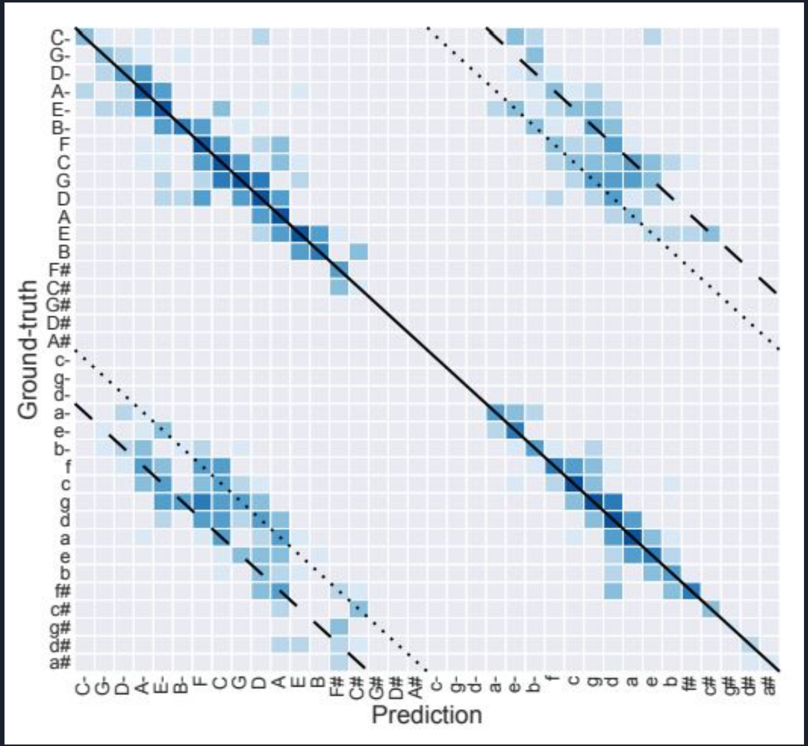
Without NADE:	78.9 %
With NADE:	99.0 %

Key Oracle

NADE has 6 outputs (including the root). What happens if an oracle gives us perfect knowledge of the first output, the key?

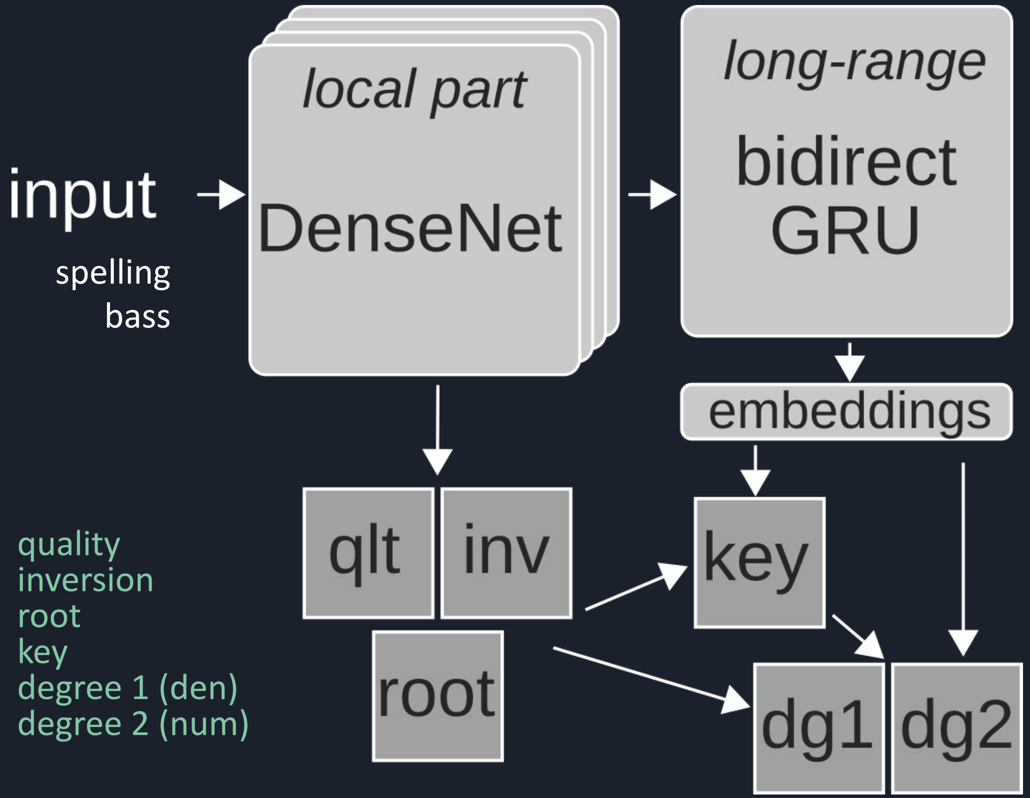
full output (1 to 6):	50.1 %
w/o key (2 to 6):	52.1 %
key oracle (1 + 2 to 6):	60.3 %

Conclusions — Analysis of the results



solid line: correct prediction
dotted lines: parallel minor
dashed lines: relative minor

Neural network architecture



local weights	33,004
long-range weights	43,392
other weights	17,511
total weights	93,907
epochs (early stopping)	42
training time (hh:mm)	2:57

Results

	Key	Degree	Quality	Inversion	RN	Tonicised	dim7
ConvGRU + PSb + global	**82.9**	**68.3**	**76.6**	**72.0**	**42.8**	24.3	**32.2**
Chen and Su (2019)	78.4	65.1	74.6	62.1		**68.2**	
Chen and Su (2018)	66.7	51.8	60.6	59.1	25.7	4.0	
Local model	67.0						

Accuracy in % on the labels

… and our model deals correctly with pitch spelling …

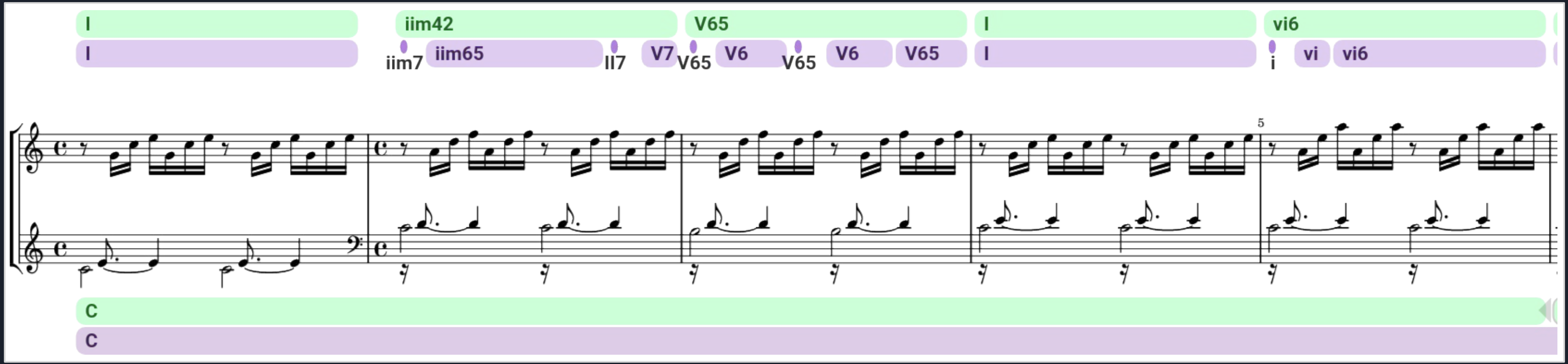
		Key	Degree	Quality	Inversion	RN	Tonicised	dim7
ConvGRU + PSb + global		**82.9**	**68.3**	**76.6**	**72.0**	**42.8**	**24.3**	**32.2**
ConvGRU	12	**81.9**	**67.4**	**74.6**	**67.9**	**37.8**	**24.7**	**32.0**
ConvDil	12	−2.4	−1.8	−0.8	−0.5	−1.7	−4.1	−4.0
PoolGRU	6	−2.3	−3.0	−1.6	−1.8	−4.1	−9.5	−4.9
bass	10	**80.8**	**66.6**	**74.3**	**70.1**	**39.2**	**22.7**	**30.4**
full	10	−0.7	−0.9	−0.6	−3.5	−3.7	−3.0	−0.7
class	10	−0.1	−0.7	−0.1	−4.7	−4.7	−1.4	−2.3
spelling	15	**80.6**	**66.2**	**74.1**	**67.6**	**36.5**	**22.2**	**31.4**
chromatic	15	−0.3	−0.3	−0.2	−0.5	−0.4	−1.9	−4.0
global	15	80.6	**66.8**	**75.4**	66.7	36.9	21.3	30.0
local	15	**+0.3**	−0.7	−2.4	**+2.0**	**+0.2**	**+2.9**	**+0.1**

Effects of all the changes we have implemented

Conclusions — The four types of errors

1. Segmentation errors
2. Mislabeling of rare chords
3. *Alternative* readings
4. Unacceptable readings

Bach WTC I : Prelude #01 in C BWV 846

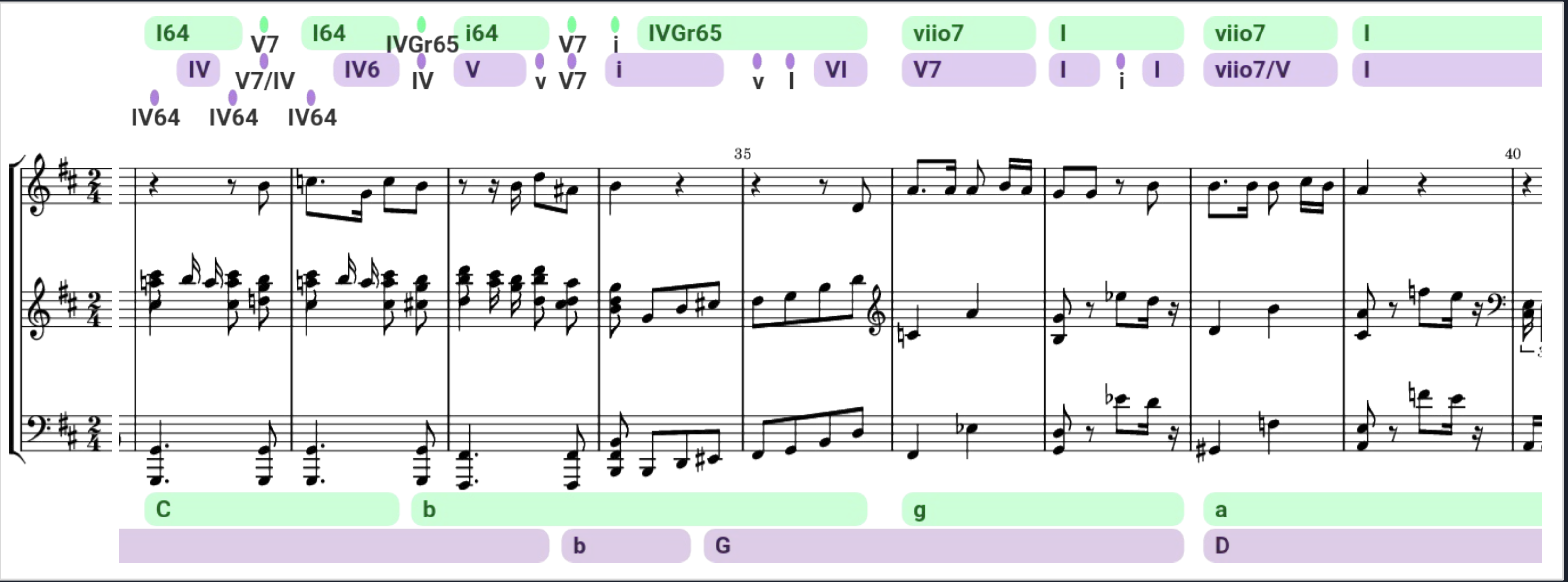


reference above
prediction below

Conclusions — The four types of errors

1. Segmentation errors
2. Mislabeling of rare chords
3. *Alternative* readings
4. Unacceptable readings

Schubert Winterreise D911 No.12 - Einsamkeit

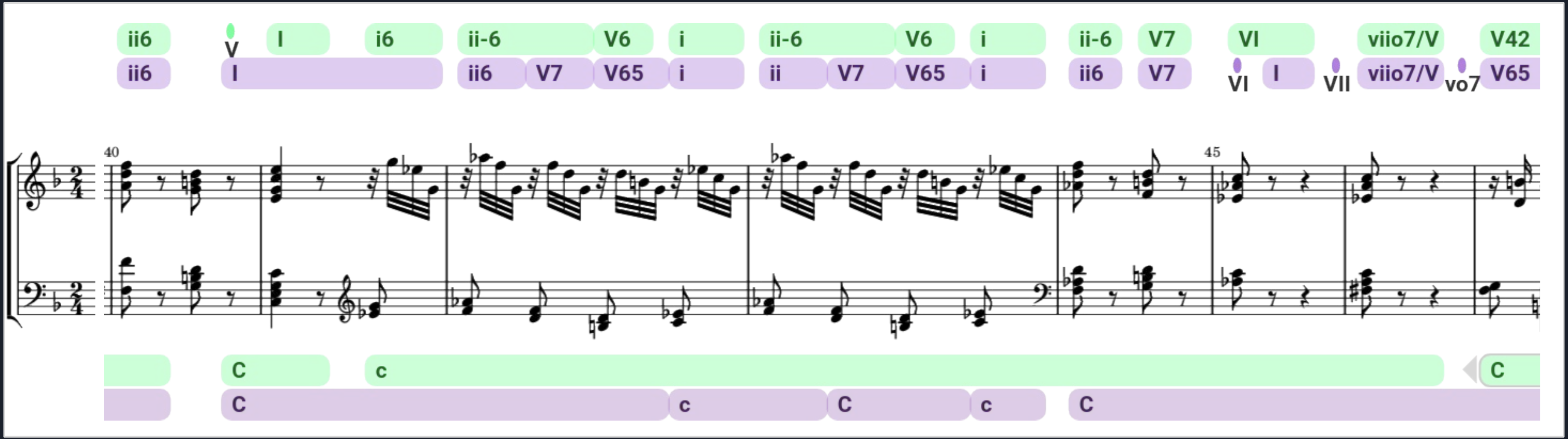


reference above
prediction below

Conclusions — The four types of errors

1. Segmentation errors
2. Mislabeling of rare chords
3. *Alternative* readings
4. Unacceptable readings

Beethoven, Piano Sonata #06 op.10 no.2



reference above
prediction below

Demo

http://roman.algomus.fr

Technical details

Frame size: 1/32 for notes, 1/8 for chords (MaxPooling layers to convert)

Conv part: 1D layers with notes as channels, context 2 quarter notes

GRU: Bidirectional, as we have knowledge of entire piece when analysing

Scores are chunked in pieces of 80 quarter notes duration