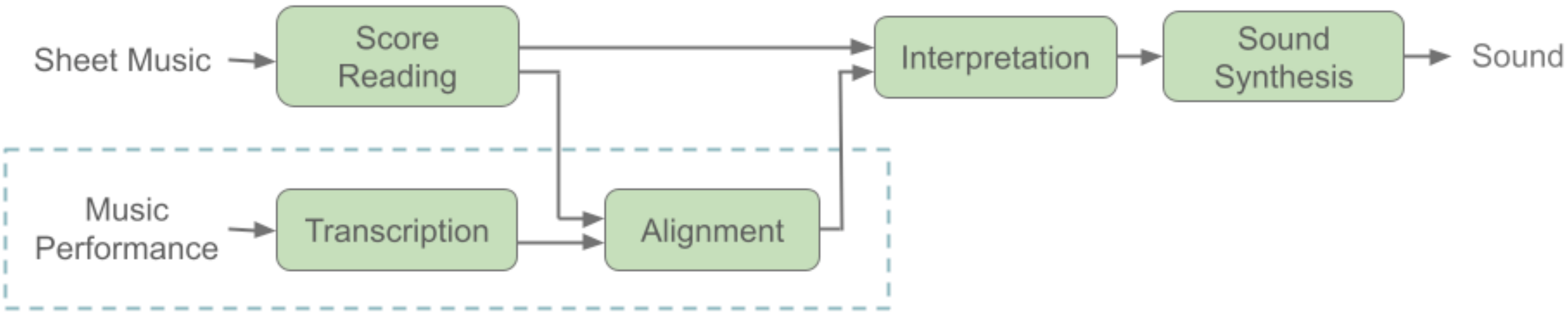
Towards a Generalized View of Music Interpretation Applied to Human-AI Collaboration

2024-11-20 Kit Armstrong

Background



Sheet Music
Score Reading
Interpretation
Sound Synthesis
Sound
Music Performance
Transcription
Alignment

Background

- *f*: Score → Interpretation
- Given our current scope, we can define:
  - Score = list of **abstract** note events + annotations
    - Note events = [on/off, pitch, score position, part]
    - Annotations = expressions, “this is the main part”, etc.
  - Interpretation = list of **realized** note events

    in MIDI environment: [on/off, pitch, **time**, **velocity**]

Background

- f: Score → Interpretation
- Why is this important?
  - Emotion in music comes from interpretation, not from score.
    - Corollary: a score can be mapped to diverse interpretations with diverse emotions.
  - Classical music is about making new interpretations from known scores.
  - Good interpretation makes music “listenable”.
    So far, computers are not good at interpretation.

Background

- f: Score $\rightarrow$ Interpretation

  {abstract note events} $\rightarrow$ {realized note events}

- Every event is mapped to exactly one event, so we can use an index to track events.

Background

- f: Score → Interpretation

  {abstract note events} → {realized note events}
- When dealing with interpretation, it's useful to still have the abstract information.

  So we consider the space of
  {abstract note events with their realization}

  The domain of f is sets of such events.

Background

- f: Score → Interpretation

  {abstract note events} → {realized note events}

- We write elements of the domain of f in the form: {notes}, where each note = [index, on/off, pitch, score position, part, time, velocity]

Background

- f: Score → Interpretation

  {abstract note events} → {realized note events}

- We write elements of the domain of f in the form: {notes}, where each note =
  [index, on/off, pitch, score position, part, time, velocity]

  $\underbrace{\text{index, on/off, pitch, score position, part}}_{\text{unchanged by } f}$

Score S		Interpretation I
$[1,o_1,\#_1,p_1,x_1]$		$[1,o_1,\#_1,p_1,x_1,t_1,f_1]$
$[2,o_2,\#_2,p_2,x_2]$		$[2,o_2,\#_2,p_2,x_2,t_2,f_2]$
$[3,o_3,\#_3,p_3,x_3]$		$[3,o_3,\#_3,p_3,x_3,t_3,f_3]$
$[4,o_4,\#_4,p_4,x_4]$	$\longrightarrow$	$[4,o_4,\#_4,p_4,x_4,t_4,f_4]$
$[5,o_5,\#_5,p_5,x_5]$		$[5,o_5,\#_5,p_5,x_5,t_5,f_5]$
$\ldots$		$\ldots$
$[i,o_i,\#_i,p_i,x_i]$		$[i,o_i,\#_i,p_i,x_i,t_i,f_i]$

Example

- EME33

  MIDI-to-MIDI Score Alignment

  Feature extraction

  Model

  MIDI Performance Construction

- MIDI-to-MIDI Score Alignment: create *S*

Example

- EME33

  MIDI-to-MIDI Score Alignment
      Feature extraction
          Model
              MIDI Performance Construction

- Feature extraction & Model:

  LSTM: $\{\#_i\} \rightarrow \{f_i\}$

  LSTM: $\{p_i\text{-}p_j \mid o_i = o_j = \text{“on”}\} \rightarrow \{IOI_i\}$

  LSTM: $\{p_i\text{-}p_j \mid \#_i = \#_j\} \rightarrow \{Dur_i\}$

Example

- EME33

  MIDI-to-MIDI Score Alignment

  Feature extraction

  Model

  MIDI Performance Construction

- MIDI Performance Construction

  $\{IOI_i\}, \{Dur_i\} \rightarrow \{t_i\}$

  $\{t_i\}, \{f_i\}$ create I

Partially fixed interpretation

- A slightly different task
- Often some entries of I are already known
- In essence, putting restrictions on the output of f

Partially fixed interpretation

- Accompaniment or collaboration are examples

- x_i is chosen from 2 options:
  {part1, part2} or {input, output}
- t_i, f_i are known iff x_i = “input”

Partially fixed interpretation

- Real-time accompaniment

- x_i is chosen from 2 options:
  {part1, part2} or {input, output}
- t_i, f_i are known iff x_i = "input" and t_i < current_time

- Task: fill in t_i, f_i for i such that x_i = "output"

  !! must satisfy $t_i \geq$ current_time

Partially fixed interpretation

- Real-time accompaniment
- Task: fill in t_i, f_i for i such that x_i = "output"
  !! must satisfy $t_i \geq$ current_time
- Objective: we want *I*[:i] ("first i notes of *I*") to be identical to the output of a "reasonable" musician *f*
  - Note: when indexing, we can ensure $i \leq j$ iff $p_i \leq p_j$
  - In a reasonable interpretation, $p_i \leq p_j$ implies $t_i \leq t_j$
  - So *I*[:i] is well-defined.

Synchronization approach

- Split *S* into *T* + *U* according to x
  - *T* = input, *U* = output
- Assume: in order to be reasonable, the relationship between p and t in *T* + *U* must be "differentiable". Discrete world – so we say "locally approx. linear".
  - ∴ the same linear relation applies in *U* as in *T*
- Assume that *T*[t < current_time] is known.
- We can figure out t_i for any given p_i of a note in *U* by using the (t,p) values in *T*

Synchronization approach

- Our Kuramoto model:
- For i in *T*, j in *U*, we use $\{p_i\},\{t_i\},\{p_j\}$ to find $t_{j'}$* where j' is the next index in *U*
- $t_{j'}$ = max($t_{j'}$*,current_time)

Synchronization approach

- Maezawa's model:
- Assume that in *U*,
  $t = xp+y$ for non-constant coefficients $x(p),y(p)$.
- Suppose $p_i = p_j$ for some i in *T*, j in *U*, and that at current time, t_i and t_j are both already known.
- Define asynchrony $a(p_i) = t_j - t_i$
- Update $x(p_{i'}) \leftarrow x(p_i) - \beta_i * a(p_i)$
  $y(p_{i'}) \leftarrow y(p_i) - \alpha_i * y(p_i)$
  where i' is the next index

Synchronization approach

- Maezawa's model:
- Update $x(p_{i'}) \leftarrow x(p_i) - \beta_i * a(p_i)$

  $y(p_{i'}) \leftarrow y(p_i) - \alpha_i * y(p_i)$

  where i' is the next index

  and $\{\alpha_i\},\{\beta_i\}$ are given by score annotations.

  - Concept: α_i,β_i small means:

    "for note i, you are the main part so don't adjust."

Linear extrapolation approach

- No prima facie splitting of S
- Assume that the attributes of each note are linear combinations of the attributes of its surrounding notes.
- At current time, some of these attributes are known and some are not.
- Perform linear regression on those which are known to find those which are unknown.

Linear extrapolation approach

- Focus on the unknowns we are tasked with finding
- Finally, force "causality" conditions:
  - any previously unknown t_i must be $\geq$ current_time
  - $p_i \leq p_j$ implies $t_i \leq t_j$

Linear extrapolation approach

- We can extend the model by assigning weights
  - "Musical concept": some notes are more important, some pairs of notes are more closely related.
  - These weights can be learned annotations.
  - They form a "relevance matrix" W
    - Each note has features o,#,p,t,f
    - So for N note events in the score, there are 5N features
    - W is 5Nx5N in the most general formulation

Architecture

- Global arrays
  “inputinterpretation” & “outputinterpretation”
  - Both known and unknown entries
- Every time new information comes, create therefrom a temporary array “conjectures” covering unknown output notes
- When a conjecture's time arrives, it goes into “outputinterpretation”