Evaluating musical score difference: a two-level comparison
Francesco Foscarin, Raphaël Fournier-S’Niehotta, Philippe Rigaux, Florent Jacquemard

To cite this version:

HAL Id: hal-01989029
https://inria.hal.science/hal-01989029
Submitted on 8 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
EVALUATING MUSICAL SCORE DIFFERENCE
A TWO-LEVEL COMPARISON

Francesco Foscarin¹, Raphaël Fournier-S’niehotta², Philippe Rigaux² and Florent Jaquemard²
¹Vertigo, CNAM Paris, France ²INRIA Paris, France

Goal: Have a diff tool for music scores, similar to the Unix diff utility

Why:
• Collaborative score editing
• Music transcription evaluation
• OMR evaluation

Issues:
• the line structure of a XML file does not reflect the its musical structure \(\rightarrow\) not meaningful to apply a text \texttt{diff}\ to XML scores.
• What exactly we evaluate? Graphical content or musical content?

Our Approach

Two comparison on two different objectives:
1. Syntactic level: the graphical content of the score (beamings, tuplets, dots, ties, etc.).
2. Semantic level: the musical content of the score (notes duration and pitch).

For both cases the output is:
• a list of the modifications to transform one score into the other,
• a value \(\Delta\), that evaluate the difference between the two scores.

1 Syntactic difference
We model the graphical score with notation trees [SI]:
• Beaming Tree
• Tuplet Tree

\[
\Delta_{\text{SYN}} = \begin{cases} 0, & \text{small differences in beamings, noteheads, ties and tuplets} \\ 32, & \text{all the score is different because of the longer C in the first bar} \end{cases}
\]

• Find Least Common Substring (LCS) between trees hash for each voice.
• Equivalent to compute an edit distance with only insertion and deletion operations.

2 Semantic difference
We model the score content with timelines (similar to MIDI files).
From each timeline we:
1. build a sequence of couples \(<\text{pitch}, \text{duration}>\).
2. Compute an edit distance with the usual operations (insert, delete and update) [4]

This difference is feature-based:
• pitch only,
• durations only,
• both features.

\[
\Delta_{\text{SEM}} = \alpha \cdot \#\text{INS} + \beta \cdot \#\text{DEL} + \gamma \cdot \text{#events}
\]

where \(\alpha, \beta, \gamma\) are user-defined cost/weight values.

References

Research supported by ANR France-Quebec MUNIR Project.

Implementation
Partial implementation in the digital music library NEUMA (neuma.huma-num.fr/).
Two REST API:
• Input: 2 midi files
• Input: 2 music-xml or MEI scores

A graphical representation with Verovio web library.

Example of the semantic difference computation and visualization (Verovio) on two monophonic scores. The edit operations are noted by different figures: square (insert), triangle (update), rhombus (delete).