On the Impact of the Music Encoding for Optical Music Recognition

María Alfaro-Contreras, Jorge Calvo-Zaragoza, José M. Iñesta, David Rizo

Music Encoding Conference 2019, Vienna (Austria)

Introduction

Optical music recognition (OMR) is the field of research that studies how to make computers read musical notation (Bainbridge and Bell 2001). Since music typesetting is a tedious and expensive process, OMR represents a key element to efficiently convert the large number of existing written musical sources into a codified format that allows for its computational process.

Recent advances in Machine Learning — namely Deep Learning (DL) — which have achieved great results in similar tasks such as text recognition, allow us to be optimistic about developing more accurate OMR systems. One of the current trends in these fields is the use of holistic systems. That is, systems that face the process in a single stage (end-to-end approaches), without taking into account the constituent parts of the documents. To develop these approaches, only training pairs are needed, consisting of problem images, together with their corresponding transcript solutions (Wel and Ullrich 2017; Calvo-Zaragoza and Rizo 2018; Baró, Riba, and Fornés 2018).

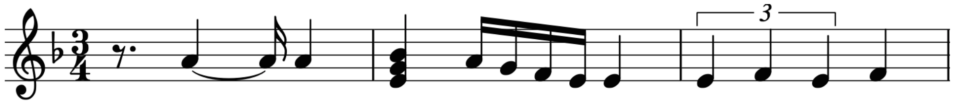
For design reasons, these models, based on recurrent neural networks, can only formulate the output of the system as one-dimensional sequences. This is straightforward for text, since it is mainly composed of character sequences. However, its application for musical notation is not so straightforward due to the presence of different elements sharing horizontal position. The vertical distribution of these elements disrupts the linear flow of the time line, that is not trivial to codify and can cause important difficulties in the performance of the classification systems that make use of the temporal relationships of the recognized elements.

XML-based music formats, such as MEI, are focused on how the score has to be coded. This makes the code plenty of irrelevant marks for the system to generate when it is recognizing the score content (mainly which symbols and where are they in the score). For that, we have designed a language to represent an appropriate OMR output, based on serializing the music symbols found in a staff. The sequential nature of music reading must be compatible with the representation of the vertical alignments of some symbols. In addition, this representation has to be easy to generate by the system, which analyzes the input sequentially and produces a linear series of symbols.

Coding proposals

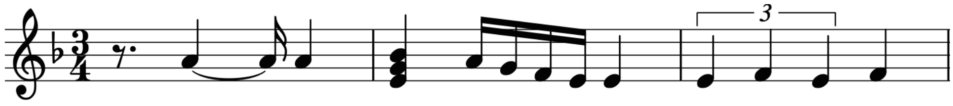
A sequential encoding representing graphical symbols in single staff scores was introduced for the task (Calvo-Zaragoza and Rizo 2018), however it does not deal with complex situations such as chords or triplets. Our current research extends that work and involves the study of four different deterministic, unambiguous and serialized representations to encode this kind of scenarios:

- **Remain-at-position character coding**: when transcribing the score, the different musical symbols are separated by a blank space except when they are in the same horizontal position. In that case they are separated by a slash, "/". This acts as a remain-at-position character, meaning that the system does not advance forward and it has to advance upwards (see Fig. 1). This behaviour is similar to the backspace of the typewriters. The carriage advances after typing and if we want to align two symbols we need to keep the carriage in a fixed position (by moving it back one position).
- **Advance position character coding**: this type of codification uses the + sign to tell the system to advance forward. This way, when the + sign is missing, the output does not move forward and a vertical distribution is being coded (see Fig. 2).



clef.G:L2 accidental.flat:L3 digit.4:L2/digit.3:L4 rest.eighth:L3 dot:S3 slur.start:S2/note.quarter:S2 slur.end:S2/note.sixteenth:S2 note.quarter:S2 verticalLine:L1 note.quarter:L1/note.quarter:L2/ note.quarter:L3 note.beamedRight2:S2 note.beamedBoth2:L2 note.beamedBoth2:S1 note.beamedLeft2:L1 note.quarter:L1 verticalLine:L1 note.quarter:L1/bracket.start-S6 note.quarter:S1/digit.3-S6 note.quarter:L1/bracket.end-S6 note.quarter:S1 verticalLine:L1

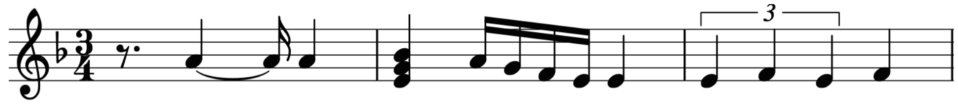
Figure 1: Musical excerpt and its transcription using remain-at-position character coding.



clef.G:L2+accidental.flat:L3+digit.4:L2 digit.3:L4+rest.eighth:L3+dot:S3+slur.start:S2 note.quarter:S2 slur.end:S2 note.sixteenth:S2+note.quarter:S2+verticalLine:L1+note.quarter:L1 note.quarter:L2 note.quarter:L3+note.beamedRight2:S2+note.beamedBoth2:L2+note.beamedBoth2:S1 note.beamedLeft2:L1+note.quarter:L1+verticalLine:L1+note.quarter:L1 bracket.start-S6+note.quarter:S1 digit.3-S6+note.quarter:L1 bracket.end-S6+note.quarter:S1+verticalLine:L1

Figure 2: Musical extract and its transcription using advance position character codification.

- **Parenthesized coding**: when a vertical distribution appears in the score, the system outputs a parenthesized structure, like **vertical.start musical_symbol musical_symbol vertical.end** (see Fig. 3).



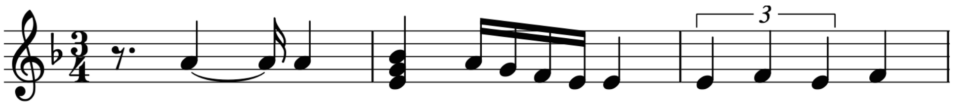
clef.G:L2 accidental.flat:L3 vertical.start digit.4:L2 digit.3:L4 vertical.end rest.eighth:L3 dot:S3 vertical.start slur.start:S2 note.quarter:S2 vertical.end vertical.start slur.end:S2 note.sixteenth:S2 vertical.end note.quarter:S2 verticalLine:L1 vertical.start note.quarter:L1 note.quarter:L2 note.quarter:L3 vertical.end note.beamedRight2:S2 note.beamedBoth2:L2 note.beamedBoth2:S1 note.beamedLeft2:L1 note.quarter:L1 verticalLine:L1 vertical.start note.quarter:L1 bracket.start-S6 vertical.end vertical.start note.quarter:S1 digit.3-S6 vertical.end vertical.start note.quarter:L1 bracket.end-S6 vertical.end note.quarter:S1 verticalLine:L1

Figure 3: Musical extract and its transcription using parenthesised codification.

- **Verbose coding**: this last codification is a combination of the two first ones. It uses the + sign as the advance position character to indicate that the system has to move forward and the / sign as the remain-at-position character to indicate that the system has to advance upwards (see Fig. 4).

Experiments and conclusions

To evaluate the impact of the musical encoding on the ability for the OMR system to produce an accurate output, four corpora of 2 000 scores, each consisting of a single staff line, have been generated using a system for automatic generation of labeled data for OMR research. Each corpus will be used to train a music score recognition algorithm based on a end-to-end neural network (Calvo-Zaragoza and Rizo 2018). In this case, each sample will be a pair composed of an image (the rendered staff) and its corresponding representation with the format imposed by one of the four musical encodings proposed in this research. These training pairs are like those shown in the



clef.G:L2+accidental.flat:L3+digit.4:L2/digit.3:L4+rest.eighth:L3+dot:S3+slur.start:S2/note.quarter:S2+slur.end:S2/note.sixteenth:S2+note.quarter:S2+verticalLine:L1+note.quarter:L1/note.quarter:L2/note.quarter:L3+note.beamedRight2:S2+note.beamedBoth2:L2+note.beamedBoth2:S1+note.beamedLeft2:L1+note.quarter:L1+verticalLine:L1+note.quarter:L1/bracket.start-S6+note.quarter:S1/digit.3-S6+note.quarter:L1/bracket.end-S6+note.quarter:S1+verticalLine:L1

Figure 4: Musical extract and its transcription using the verbose codification.

figures from 1 to 4. Our experiments involve using the trained systems for transcribing the content of a test set of 500 music score images presented to the system as independent (unseen) images.

We use the normalized average of editing errors as metric, equivalent to the Word Error Rate (WER) in text or speech recognition. This evaluation measure calculates the minimum number of corrections (insertions, deletions and substitutions of one word for another) necessary to match two sentences. Therefore, establishing a reference with the experiment performed, the WER is calculated between the score recognized by the OMR system and the reference score.

As illustrated in Fig. 5, our serialized ways of encoding the music content prove to be appropriate for DL-based OMR, as the learning process is successful and low WER figures are eventually attained. In addition, it is shown that the choice of the encoding have some impact on the convergence of the training process (see Fig. 5a), as the *advance position character* coding depicts a steeper learning curve. The choice also affects the lower bound of the WER that can be attained (see Fig. 5b), which almost directly correlates with the tendency of the learning curves. These facts reinforce our initial claim that the encoding of the output for OMR deserves further consideration within the end-to-end DL paradigm.

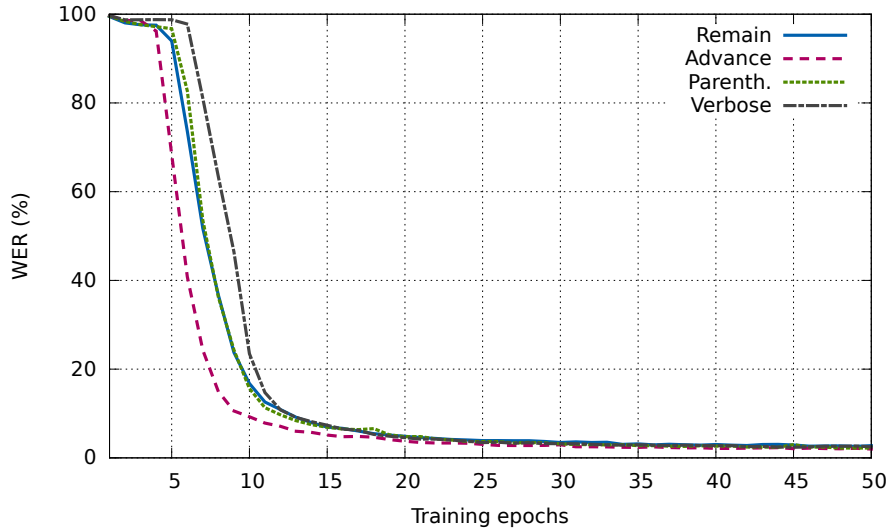
References

Bainbridge, David, and Tim Bell. 2001. "The Challenge of Optical Music Recognition." *Computers and the Humanities* 35 (2): 95–121.

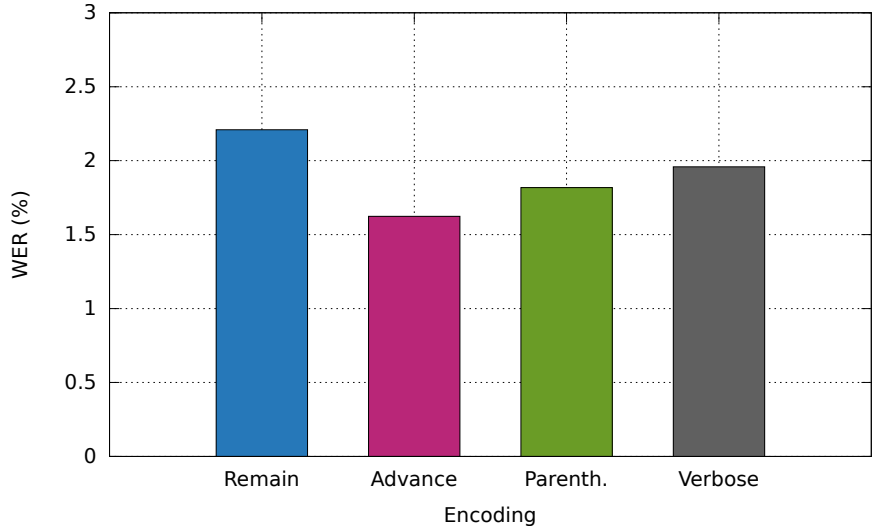
Baró, Arnau, Pau Riba, and Alicia Fornés. 2018. "A Starting Point for Handwritten Music Recognition." In *1st International Workshop on Reading Music Systems,* 5–6. Paris, France.

Calvo-Zaragoza, Jorge, and David Rizo. 2018. "End-to-End Neural Optical Music Recognition of Monophonic Scores." *Applied Sciences,* no. 4: 606. ISSN: 2076–3417.

Wel, Eelco van der, and Karen Ullrich. 2017. "Optical Music Recognition with Convolutional Sequence-to-Sequence Models." In *18th International Society for Music Information Retrieval Conference,* 731–737. Suzhou, China.



(a) Convergence analysis: accuracy over the test set with respect to the training epoch of the deep neural network.



(b) Error-bound analysis: best accuracy attained by each coding over the test set.

Figure 5: Experimental results with in terms of WER with respect to the different proposed encodings. **Remain** stands for *remain-at-position character* coding, **Advance** stands for *advance position character*, **Parenth.** stands for *parenthesized* coding, and **Verbose** stands for *verbose* coding.