International Journal on Document Analysis and Recognition (IJDAR) (2024) 27:379–393
https://doi.org/10.1007/s10032-024-00485-8

SPECIAL ISSUE PAPER

A unified representation framework for the evaluation of Optical Music Recognition systems

Pau Torras[1,2] · Sanket Biswas[2] · Alicia Fornés[2]

Received: 13 November 2023 / Revised: 28 May 2024 / Accepted: 30 May 2024 / Published online: 24 July 2024

Abstract
Modern-day Optical Music Recognition (OMR) is a fairly fragmented field. Most OMR approaches use datasets that are independent and incompatible between each other, making it difficult to both combine them and compare recognition systems built upon them. In this paper we identify the need of a common music representation language and propose the Music Tree Notation format, with the idea to construct a common endpoint for OMR research that allows coordination, reuse of technology and fair evaluation of community efforts. This format represents music as a set of primitives that group together into higher-abstraction nodes, a compromise between the expression of fully graph-based and sequential notation formats. We have also developed a specific set of OMR metrics and a typeset score dataset as a proof of concept of this idea.

Keywords Optical Music Recognition · Representation · Evaluation · Datasets · Computer vision

1 Introduction

Written music has been part of the cultural heritage of humankind for many centuries. From neumes as old as a thousand years to current-time Western notation scores, people have built methods to make this most ephemeral of arts persistent over time. Rivers of ink have been imprinted on libraries worth of paper to preserve music along the ages, and much of it has indeed survived until today.

It is therefore unsurprising that, given the sheer volume of notably interesting (and oftentimes, forgotten) pieces of music that have been endowed to our current generations, scholars have turned their attention to computers to aid them in the endeavour of preserving and analysing them. The task that concerns us in this work is Optical Music Recognition (OMR), which is fundamental step for the computational analysis of written scores; converting images or scans of music into a defined format a computer can process [18].

Modern-day OMR has been greatly enriched with the advent of the powerful deep-learning techniques developed during the last decade [18, 43]. Nevertheless, the field itself remains fragmented, with few researchers fully devoted to it and each of them developing their unique point of view and methodology [18, 35]. This is particularly evident when analysing the available datasets [37], as can be seen in Table 1; most of them are restricted to specific steps or approaches [14, 39, 53] and almost none of them are compatible with each other (the most notable exception being the DoReMi dataset introduced in 2021 [48], which incorporates its ground truth in multiple formats).

Another point of disagreement among OMR researchers is the matter of the evaluation of models [18, 29, 30, 34], which is but the visible consequence of many long-standing issues—some related to the inherent difficulty of the field (for an example, see Fig. 1), but others related to the diversity of plausible approaches to music recognition.

Evaluation of OMR models is currently performed on a per-methodology basis [18, 43]. If one considers the goal of OMR to produce a computational representation of music—whichever that final representation might be[1]—one would

✉ Pau Torras
Pau.Torras@uab.cat

Sanket Biswas
sbiswas@cvc.uab.cat

Alicia Fornés
afornes@cvc.uab.es

1 Autonomous University of Barcelona, Barcelona, Spain

2 Computer Vision Center, Computer Science Department, Universitat Autònoma de Barcelona, Cerdanyola del Vallès 08193, Spain

[1] Byrd et al. [11] collected a very extensive and detailed set of requirements for a music notation system that is suitable for both score

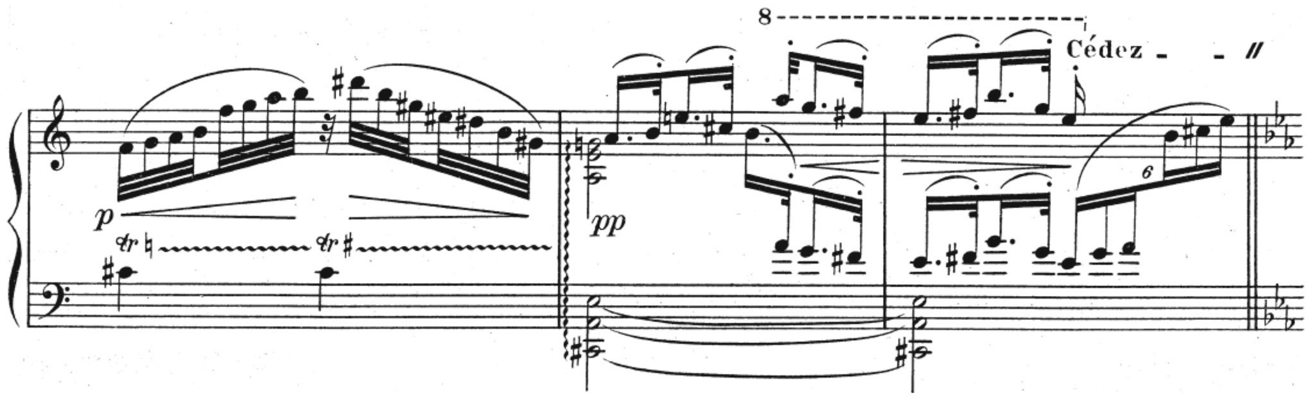

Fig. 1 Fragment of a scan of a 1910 edition of Claude Debussy's *La Danse de Puck* from the first volume of his *Préludes*. In this fragment, many of the reasons why modelling OMR scores is difficult can be seen—a two-staff system with voices that move along both of them, certain areas of the score accumulating many small symbols, many independent voices sounding in unison, elements spanning multiple measures, chords, etc

expect the metrics to obey a criterion of either some aspect of music understanding or quantifying the correctness of the final representation. Instead, most metrics revolve around measuring the fidelity of intermediate representations devoid of musical meaning (e.g., sequences of tokens or bounding boxes), disregarding the full reconstruction of music scores altogether. Moreover, the benchmarks these evaluation metrics are computed upon are rarely widespread, with each methodology using their own. Commercial OMR systems have also been very vague about the specifics of their capacity.

With the aim of setting a step forward towards unifying the efforts of the community, we propose a way to set a shared framework for OMR. We believe **the first step is to set a target final representation that accommodates a significant amount of use cases within Common Western Music Notation (CWMN)**. This is the music notation system that has been employed in Europe from the early 18th century until today.

Focusing on CWMN specifically instead of the broad range of western notations related to it is not an arbitrary choice. While CWMN did indeed evolve from Mensural notation, which was employed from the 13th to the 17th century, and consequentially shares a considerable amount of graphical constructs, this system has semantic concepts that make them quite distinct. For instance, the concept of beat as we know it today was developed during the first half of the 18th century; before that, the rules that described the relation of note types with their divisions implied different abstractions and note types that are not compatible with those of modern notation. Moreover, there are constructs such as ligatures that have very unique properties but have disappeared from CWMN. In this regard, given that each notation system has a distinct nature and given the necessity for OMR systems in CWMN, we believe it is best to commit to this specific notation alone.

Once this target representation is chosen, a way of easily working with it has to be developed. Currently used encoding formats are quite exhaustive, verbose and option-filled, which allows expressing the same score in a variety of ways—something that makes them impractical for both recognition and direct comparison. Thus, we propose an intermediate representation that translates directly to the desired output and offers analysis, conversion and evaluation tools regardless of the underlying approach. We propose doing so through a notation based on the idea of building a tree structure; leaves are the objects in the music score and intermediate nodes are their combination. The idea is exploiting the structural advantages of graph-based approaches over sequence-based ones while restricting the algorithmic complexity of working with arbitrary graphs. Incorporating a shared notation format is also an opportunity for unifying all existing datasets so that they can be both combined and compared on equal grounds, regardless of the methodology they were originally designed to work with.

The contribution of this work can be summarised by the following claims:

- We try to bridge the gap between the different benchmark suites in OMR literature with a universal tree-based notation format designed to represent musical scores at the graphical level.[2]

engraving and OMR, but to the best of our knowledge there is no recognition system that formally or explicitly applies them.

[2] Repository of the project: https://github.com/CVC-DAG/comref-converter.

- We also present an evaluation toolkit which aims towards unify existing benchmark OMR tasks for fairer comparison.
- We have produced a typeset dataset using public domain works with permissive licenses.[3]

This paper is structured as follows. In Sect. 2 some definitions for recurrent topics in the article are presented. Then, Sect. 3 justifies the necessity for an OMR evaluation framework and debates about how that framework should be structured. In Sect. 4, our proposed notation format is presented and described in detail, whereas in Sect. 5 we describe the evaluation metrics to be employed with it. In Sect. 6 we introduce the proof-of-concept dataset on typeset scores. Finally, Sect. 7 closes the paper with some final remarks and future work.

2 Terminology

In this section we shall address some specific recurrent terms for the sake of clarity and cohesiveness of the text. Throughout this paper we shall adhere to the nomenclature and definitions found in Calvo-Zaragoza et al.'s [18].

By *structured output* we refer to an output encoding format that allows the replication of the input score exhaustively and unambiguously. This is in contrast to formats such as MIDI, that only encode certain aspects of music (in this case, playback information).

Another relevant point is the definition of an *end-to-end* OMR model. Modern day literature employs this term frequently, but it must be noted that it is used with a slightly different meaning than the term itself actually conveys. The most widely recognised definition of the OMR pipeline is perhaps that of the review authored by Rebelo et al. [43], which states four distinct steps:

1. Image preprocessing.
2. Recognition of Musical Symbols.
3. Reconstruction of Musical Information.
4. Construction of a Musical Notation model.

Most end-to-end approaches do not actually perform all four steps; rather, these models stop at step 3 with a notation that encodes most of the score's information directly or indirectly, but cannot be directly engraved (sometimes requiring an additional processing step before all music semantics can be successfully generated). Whenever we address end-to-end approaches, we refer to these. In this work we mostly address *offline* use cases of OMR—that is, recognition of raster images of scores. Nevertheless, we believe some of our ideas are equally applicable to online use cases [13].

3 The necessity for an OMR framework

In this section we comment on some of the main issues of the field of OMR and provide arguments as to why having a unified OMR framework is useful and necessary.

3.1 OMR needs a standardised output

Most OMR research today does not tackle the problem of converting a model's output to a symbolic representation that can be engraved into a proper score. In their review of the field in 2021, Calvo-Zaragoza et al. [18] go as far as claiming that no research OMR system that they know of is actually capable of performing such a conversion,[4] and since its publication we have only acknowledged one system that can [49]. One of their main justifications for this is the fact that most structured music engraving formats are either not mature enough to cover all possible use cases or impractical for OMR tasks due to their complexity.

There are currently two main choices for music score exchange formats: MusicXML [26], devised by Recordare and now managed by a W3C committee and the Music Encoding Initiative (MEI), designed by Perry Roland [47]. Other formats such as Lilypond [51] (an open-source text-based music engraving software) or Humdrum **kern [32] (a notation format for the automatic musical analysis tools in the Humdrum suite) have been used for OMR as they allow engraving scores very succinctly. Nevertheless, these formats are not designed to represent scores precisely and unambiguously—beaming groups are not explicitly modelled, multiple representations of the same score through permutations are allowed (although **kern has been extended to fix this [46]), ornaments, expression signs, dynamics and other non-structural elements are only partially supported, etc. As a consequence, these formats are not suited to engrave exact replicas of any music score.

MusicXML and MEI are somewhat similar in their base design ideas, with the former being more widely used within music score repository platforms and engraving software and the latter in archival settings. Thus, we believe it is important to make all OMR applications interoperable with these formats by design. Doing so makes it easier to collect data for OMR applications and also simplifies deploying OMR systems. Having a well-defined target representation has the additional side-effect of fixing the requirements for what is

[3] Link to the dataset https://datasets.cvc.uab.cat/comref/comref.zip.

[4] There are however commercial OMR systems [9] that can perform this step.

expected from an OMR system, encouraging joining efforts within the community.

Since working with these formats directly can be difficult, it would be interesting to have a different intermediate representation that acknowledges the requirements of OMR but can convert to other formats when required.

3.2 Trade-offs for structured representations of music for OMR

To this day, there is no straightforward answer as to what is the best basic data structure to represent music for OMR purposes. There are various approaches being explored in the community, each having their own set of trade-offs.

The most flexible—and perhaps the most logical—computational representation of written music is most likely a graph, since it allows expressing arbitrary relationships between primitives and symbols. There exists a very widely known notation format based on this idea called Music Notation Graph (MuNG) introduced for the MUSCIMA++ Dataset [30]. Some works have indeed succeeded in being able to reconstruct music structure from these graphs, such as Baró et al.'s Musigraph [5] or Garrido-Muñoz et al.'s Convolutional Recurrent Neural Networks [25]. Nevertheless, to the best of our knowledge, there is no straightforward way of converting this graph output back into a structured encoding. Moreover, the cost of evaluation on graphs is high due to the complexity of matching them.

At the other end of the spectrum lies the sequential music model, currently employed in some form or another by most end-to-end OMR approaches [1, 2, 4, 7, 7, 15, 16, 20, 45, 46, 52, 56]. These methods impose a relationship between the horizontal axis of the image input and the playback time of the score. However, this forces either having to model music time semantics when inferring the structure of the score or introducing simplifications to music notation. The latter is the most common, with information such as stem directions or beaming groups being abstracted away to varying degrees.

3.3 Annotated data is scarce

One of the longest-standing problems in OMR is the lack of reliably annotated data for contexts other than recognition of typeset scores. Even in that case, the datasets are usually tailored to specific recognition techniques and rarely incorporate structured music sources (see Table 1 for more detail). Most datasets are incompatible between each other as a result.

In this sense, it would be very positive that all datasets provide an additional structured source in a common format in a way similar to [48], so that the original data can be used in contexts outside the ones initially intended. This would help develop better systems overall by having the possibility of ensembling datasets in those domains where there are many of them available.

However, there are some domains for which there is barely any data available. An example of this are historical CWMN, written from between the 17th and 20th centuries. As can be seen in Table 1, only Baró et al.'s [4] Pau Llinás dataset is dedicated to music from this period. This is caused for various reasons:

- The only realistic way of annotating a corpus of handwritten music from scratch given the tools existing today is doing so manually, which is extremely time-consuming.
- Crowdsourcing annotation of music scores is difficult because it requires trained individuals. Some music scores in this period are notoriously difficult to read.

Making it possible to incorporate structured sources opens up the possibility of using existing public domain transcriptions works of widely known authors and trying to synchronise them to the original public manuscripts (e.g., Bach [3] or Beethoven [8]) at a significantly lower cost.

3.4 Evaluation of OMR models

Evaluating OMR models fairly is important in order to be able to understand the strengths and weaknesses of each system against the rest and being able to dissect the effect of design decisions on the final output.

Unfortunately, most metrics used by the OMR community in modern day are fairly domain-specific and rarely take into account a final representation of music [34]. Authors that address OMR as an object detection problem report object detection metrics, whereas authors that address OMR as an image-to-sequence task rely on sequence edit distances. Consequently, there is no realistic way of comparing them since the semantics of the mistakes of the system are lost and the underlying notations are different. The few authors that do report musically-aware metrics restrict themselves to per-note pitch and/or time accuracy [7, 31].

In this direction, there is also debate regarding the specific aspects of music that should be measured. Hajič [29] argues that the best metric is one that summarises the entire notation system as one single value, reminiscent of the Symbol Error Rate (SER) metric for strings. We believe that, while such a metric is a good indicator of overall performance (and is definitely useful for benchmarking), it must not be considered on its own, as definite conclusions about the underlying models are very hard to make. Our belief is that there should be an array of metrics that are oriented at different aspects of the score that should be logged together, increasing the coverage of the system. Both cases only make sense under the assumption that comparisons are made on a standardised set of objects and primitives.

Table 1 Description of annotation formats available for OMR datasets designed for final output production (i.e., they are designed to reconstruct a full score in some form)

Publication	Dataset name	Dataset type	Boxes?	Masks?	Graph?	String?	Struct?
[30]	MUSCIMA++	Handwritten contemporary	✓		✓		
[53, 55]	DeepScores	Typeset	✓	✓			
[54]	RealScores	Scanned	✓	✓			
[48]	DoReMi	Typeset	✓	✓	✓	✓	✓
[39]	SEILS	Typeset scanned mensural				✓	✓
[4]	Baró Synthetic	Typeset				✓	
[4]	Pau Llinás	Handwritten historical				✓	
[16]	PRIMuS	Typeset				✓	✓
[14]	Camera PRIMuS	Typeset				✓	✓

Columns from left to right: the publication that spawned the dataset, the name it is usually referred to as, what type of music samples it contains and whether it contains bounding box annotations, pixel-mask annotations, notation graph (MuNG) among symbols, string-based output annotations and a final structured output (MusicXML or MEI)

3.5 The ideal scope for recognition

In this final section we discuss the effect of tackling OMR at different levels of detail.

Presently, there is no pre-defined scope to tackle the problem of OMR. Therefore researchers can choose whether to implement recognition at measure [4, 5], line [7, 14, 56] or page level [38, 44, 59]. While there are practical advantages on each perspective, there is an arguing point that should be brought to discussion given the current situation of OMR and Computer Vision as a whole.

Current state-of-the-art object detection and recognition models struggle with both small [5, 36, 57] and scale-variant symbols [58]. Music scores are quintessential examples of both phenomena: large symbols (slurs, ties, staves) coexist with very small ones (dots, fingering numbers), and writers draw symbols at their preferred relative scale without affecting their meaning—something that is particularly true for noteheads and stems—while also writing semantically-distinct symbols that change based on their size—grace notes. Additionally, the number of objects present in the input image also affects the recognition performance, particularly when objects accumulate in a limited region of the image.

Considering this, it should be possible to tackle OMR from the smallest scope possible: the measure level. Addressing higher scopes can be achieved by composing smaller semantic units. The downside is that this requires more effort to be put in the analysis of the layout of the scores, as all successive recognition steps depend on finding the measures reliably on the page. Nevertheless, it can be argued that it is an objectively simpler problem than recognition: staves and bar lines are very prominent, very ubiquitous and can easily be identified heuristically [10, 17, 19, 21, 22, 24], as opposed to many of the other musical symbols.

In this section we have provided the core ideas for a common OMR notation format and evaluation procedure. This shall be the logical foundation upon which our proposed solution, described in the following section, is built.

4 The music tree notation format

As stated in Introduction and in concordance to the necessities discussed in Sect. 3, we have designed a notation format that

- normalises the set of music primitives to be recognised,
- simplifies conversion to a final structured format,
- enables comparison of diverse OMR methods on equal grounds and
- facilitates the usage of non-OMR-specific data.

The base logic for the notation system is introduced in Sect. 4.1. The notation system is described in detail in Sect. 4.2.

4.1 Rationale

The basis to our proposal for an OMR framework for inference, score reconstruction and evaluation is the Music Tree Notation format (MTN). We believe the final goal of structured OMR is the reconstruction of the score at the visual domain. The core idea of this format is therefore to build a notation that exclusively models relationships between graphical symbols and defers inference of music semantics until a later stage. Only those high-level music concepts that are strictly required to reconstruct the score unambiguously are kept if and only if there is a direct graphical cue that allows straightforward inference.

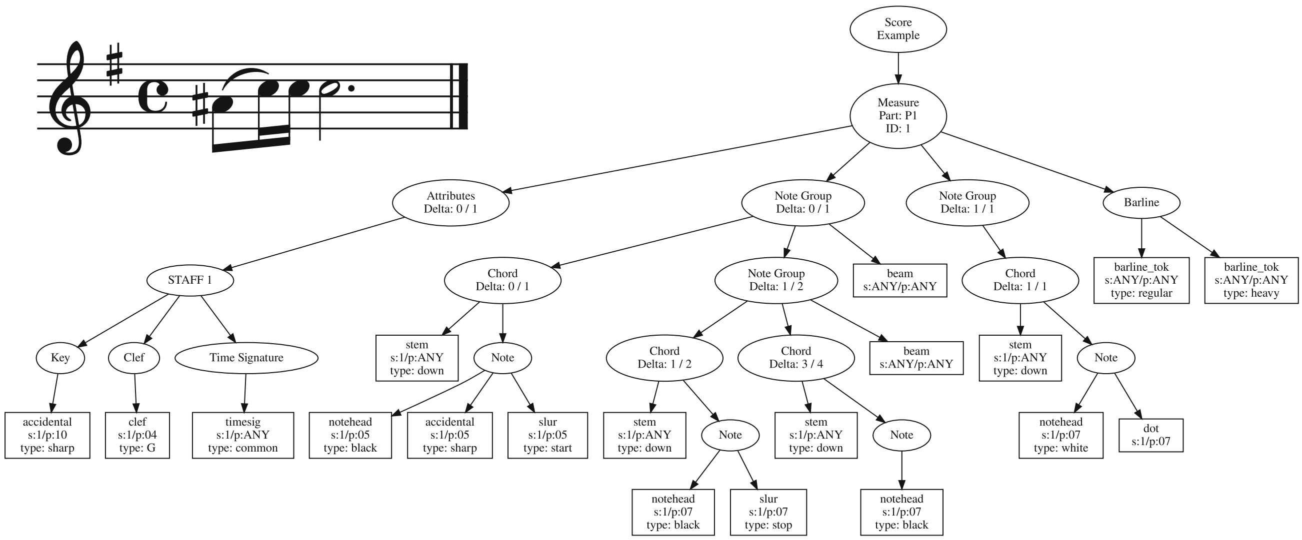

Fig. 2 Example showing a fragment of a measure in which the annotation format for attributes and staff-modifying elements is shown. Rectangular nodes represent primitives as tokens and rounded nodes are abstract elements

Score reconstruction should be possible regardless of the scope of recognition of the underlying system—page, line or measure. Nevertheless, a minimal semantic unit is still required in order to reconstruct music semantics afterwards and facilitate alignment of predictions and their groundtruth. Exploiting the assumption OMR is performed on the visual domain and therefore application of semantic-altering objects is irrelevant, we propose using the measure as the minimal unit of music on which to evaluate. Moreover, reconstruction should be possible regardless of the origin of the transcription. The format must not require localisation of symbols—only their relationships—in order to accommodate the use of end-to-end or pipelined techniques interchangeably.

MTN should act as a final target for OMR, through which to convert the score into any other high-level notation system. Employing a tree structure simplifies developing converters to new formats, which can be implemented by writing a tree traversal algorithm at linear complexity. This has the potential of facilitating OMR systems from research being deployed into real use-cases as well as enabling any source of structured data to be used in OMR applications. Therefore, MTN draws heavy inspiration from both MEI and MusicXML, but "normalises" their structure in such a way that every score can only be expressed in one way. This also allows implementation many of the already-studied evaluation requirement ideas from other authors, particularly Hajič [28] and Byrd [12].

Finally, music notation is very rich and tends to bend its own rules quite heavily. **The goal for this notation and evaluation system is to accommodate a most significant subset of CWMN**, but covering every single exception proves to be a daunting task. Byrd [12] presents some specific cases such as a spanning slur with seven inflection points that is rather complicated to model in the logical domain. Therefore, our rationale is that support for these should not be a priority.[5] We believe it is better to have some solid tools on a core subset of music rather than trying to cover every possible case with highly complex and fragile ones as this makes adoption of the format much more difficult. In this regard, the format is designed so it is very easy to add support for new types of music primitives or add new sources of information.

4.2 Description of the format

A graphical representation of a simple measure engraved in MTN can be seen in Fig. 2. The core element of this format is the Musical Primitive, a concept that is quite widespread in the OMR literature [4, 6, 14, 53] and can be defined as any of the independent structural elements that may or may not be combined together to form a semantic unit in the music score. The set of musical primitives includes all graphical elements in a score that are self-contained and require no other symbols to convey meaning (this includes rests, clefs or time signature symbols), the set of graphical elements that compose notes (noteheads, stems, flags, dots, accidentals, etc.) and other miscellaneous elements such as numbers for compound time signatures. Every primitive is given a unique work-level identifier.

[5] This specific change could potentially be modelled in MTN using the tools already in the format either by providing the bounding box of the slur (which would be rather imprecise) or its pixel mask associated to its ID in the tree.

Table 2 Definition of the main abstract elements within MTN and the overall structure of the format

Top level	Level 1	Level 2	Definition
Attributes			Any semantic change the score goes through
	Staff		Group of changes occuring on the same staff
		Clef	A change of clef in the score
		TimeSig	A change of time signature in the score
		Key	A change of key in the score
Barline			A measure separator and its attached modifiers
Direction			A single playback directive
Note Group			Group of notes tied together by at least one beam
	Note Group		*Note Groups can be nested, each owning one or more beams*
	Chord		Set of notes in unison attached to the same stem, if it exists
		Stem	Stem direction information and beams or flags attached to it
		Note	A notehead and its attached modifiers
Rest			Rest token and its attached modifiers

Each of the lowest levels of the notation contain the various final tokens. For more information, refer to the implementation of the format

These primitives associate together to form more abstract constructs. This is modelled in MTN using a tree-like structure of higher-order elements resembling an Abstract Syntax Tree (AST), which defines the set of dependencies among objects in the score. This idea emulates parsing the contents of the score using a grammar, enabling the bulk of tools and research on parsers, parser generators and AST analysis and processing to be used in the context of music. Furthermore, it is a structure that can be modelled very easily using an exchange format such as XML. The structure of the format may be found in Table 2.

Tree-based music representations for evaluation of recognition has been explored before in [23]. In this paper they also nest tuplets within the hierarchy. Nevertheless, this is not possible for any tuplet in MTN because they can extend beyond the scope of the current group and incorporate neighboring rests or other objects, which would break the tree assumption of the format. Instead, this is modelled using two start/end tokens.

There are some elements in music that break the tree-like structure assumption. These are elements that connect multiple notes together outside their local note group structure: slurs, ties, parentheses and tuplets, among others. Both MEI and MusicXML acknowledge this limitation and circumvent it through the use of identifiers. MTN is no different: it provides a unique starting and ending token for each side of the object and gives both ends the same identifier.

In order to describe the position of MTN elements, two magnitudes are used. Firstly, for every token a tuple of two integers denotes the staff the element belongs to and its position within the staff. The position is denoted counting the number of steps from the first ledger line below a staff. For those elements without a specific position (such as rests or stems), a null value is used. Secondly, for any object immediately below the class measure, an exact timing value is provided. It is measured in fractions of a quarter note from the start of the measure itself. This information is also provided for every chord in a note group even if this information can be inferred for the sake of simplifying evaluation procedures.

This musical information is necessary for two reasons. The only visual cue for aligning elements in a polyphonic setup is that, at some point, additional voices have at least one object that is perfectly aligned with one in the main voice. This is very hard to model cleanly with anything other than a graph. Moreover, the main voice of a measure may arbitrarily jump from staff to staff in a system, making timing generally impossible to resolve on a per-staff basis, as can be seen in Fig. 3. A small optimisation for sequence-based notation systems is that only the first element in a group needs timing information, as all successive chords' timings can be inferred from the note types and tuplets attached to them (or their surrounding elements).

Doubly stemmed notes (such as in the last measure of the Arabesques example in Fig. 3) can be interpreted as two notes playing at the same time. Therefore, they can be modelled using two distinct groups without needing to break the tree-like structure.

Finally, to produce unambiguous scores, a reading order of sorts must be established. We propose the following ordering criterion:

- By starting time counting from the beginning of the measure.
- By top level class:
    1. Attributes

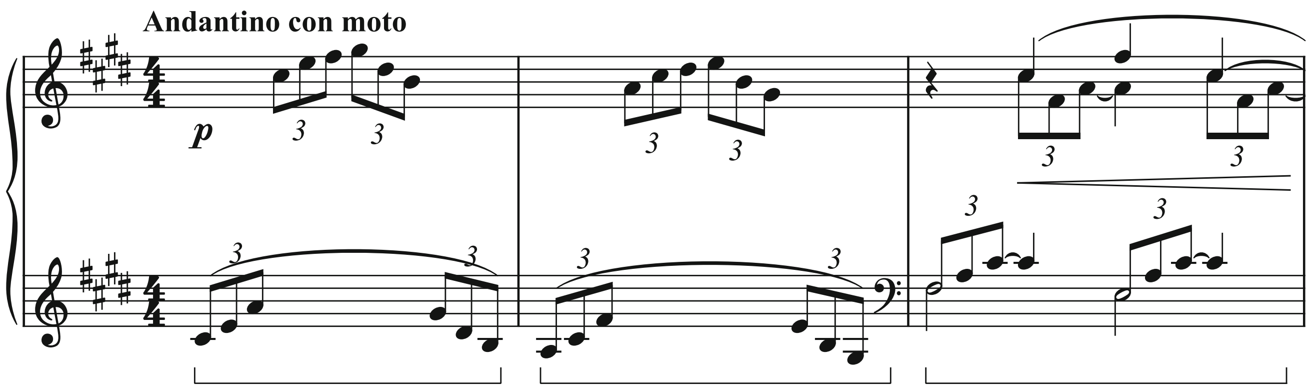

Fig. 3 The first few measures on Claude Debussy's first Arabesque. The single voice moves along both staves and is played using both hands on a piano. If one was to separate both staves, it would be impossible to extract the exact timing where each group of notes is to be played. Another interesting aspect of this piece is that many editions use implicit triplets, something that only can be inferred contextually by the fact that eighth notes are on groups of three

2. Directions
3. Rests
4. Note Groups
5. Barlines

- By staff position: first objects on upper staves and lower positions within them.
- In case of Note Groups, by direction of the first stem: first stems looking upwards.
- For anything else, token alphabetical order. This also guarantees stability of the notation if new token types are added.

4.3 Extensions

The main goal for the notation is to address symbolic-level OMR. Nevertheless, many practitioners use pipelined architectures where some partial results need to be annotated (most notably, object bounding boxes). Providing these annotations with MTN is straightforward: it is a matter of adding an "annotations" field with each token identifier linked to the desired information. This also includes some other relevant semantic cues such as voices, which we deem unnecessary for the task of score reconstruction but might be useful for some other music information retrieval tasks. A concern surrounding voice information is that its definition around music engraving software is more akin to a layer in a graphic design suite; it is mostly used as a way of circumventing engraving limitations, making voice information in files like MusicXML rather arbitrary.

The presence of text in scores can also be treated separately from the rest of music semantics. Since its interaction with the score is very situational and depends a lot on the contents of the text itself, it is very difficult to devise a set of relationships that might be useful for most scores. Therefore, the only text that is to be represented in MTN are dynamic directives, which are common and well-defined. For the rest of the text such as tempo indications or other textual remarks, we believe it is best to append a set of text boxes in a similar way as bounding boxes are provided.

The specification of the format provides ways of developing new extensions as new needs arise.

5 Evaluation metrics

We propose a set of evaluation metrics that both acknowledge the existence of multiple paradigms for OMR while also setting ways to compare any structured output equally. These metrics draw inspiration from the currently used Symbol Error Rate and some ideas from Hajič [28]. We have divided our proposed metrics as tiers depending on the abstraction level they address and the problems they can help diagnose.

5.1 Tier 0: Methodology-specific metrics

The first tier is that devoted to specific pipeline metrics before performing any kind of reconstruction into the MTN format. The idea behind this tier is that comparison of methodology-specific metrics is still useful, since it might help identify strengths or shortcomings on certain applications and help quantify the effectiveness of methods converting into MTN format (which depending on the OMR methodology might be more or less nuanced).

Since the approaches to OMR are both diverse and unique, it does not make much sense to overly specify what metrics to employ under this tier, but rather use those metrics that are already being employed. Some examples of the kinds of metrics that should be reported here include (but are not restricted to):

- For object detection methods, metrics such as mean average precision, F1 score, average intersection over union and so on.
- For end-to-end string-based methods, metrics such as symbol error rate or string parseability (whether or not the output string can be interpreted correctly with whichever grammar describes its underlying language).

5.2 Tier 1: Primitive detection

The first set of metrics addresses the presence or absence of terminals within the MTN string. The main goal is determining the capacity of models of detecting objects and analysing token co-ocurrence phenomena. This is especially important in OMR because the distribution of classes in music is extremely unbalanced: Noteheads are orders of magnitude more common than almost any other token. Understanding what tokens are missed and why is key to developing a good recognition system, since missing a single stem can alter the reconstruction of the score completely.

These metrics do not take into account structural matters, making them quick to compute. Given a set of unique terminals T, which is a subset of the full MTN vocabulary V, the multiset of predicted terminals within a sample $P = \{p_1 \cdots p_n : p_i \in T\}$ as the multiset of all terminals produced by a model and the multiset of ground truth terminals within a sample $G = \{g_1 \cdots g_m : g_i \in T\}$ as the multiset of all terminals existing in the groundtruth (both in no particular order), we define the following metrics:

- Primitive-level precision

$$precision = \frac{\|P \cap G\|}{\|P\|} \quad (1)$$

- Primitive-level recall

$$recall = \frac{\|P \cap G\|}{\|G\|} \quad (2)$$

These metrics are computed per-class for the entire dataset. In order to produce a single precision and recall measure, results are aggregated per-class using a weighted average, where the weights are the relative frequency of each token in the ground truth.

5.3 Tier 2: Structure reconstruction

This tier takes into account the structure of the produced MTN and compares it directly with that of the ground truth. A matching from ground truth elements to those present in the prediction is performed using a tree edit distance algorithm. In particular, since there is a restriction on the ordering of sibling labels, the $O\left(n^3\right)$ solution from Zhang and Sasha can be employed [60]. In practice, we use a Python implementation [33] of Pawlik et al.'s APTED algorithm [40].

Given the following operations:

- *Substitution*: Changing the label of a single node within the tree.
- *Deletion*: Removing a single node of the tree and setting its children as siblings.
- *Insertion*: Adding a new node under a parent one and setting a consecutive subsequence of its siblings as children.

Given a predicted tree and a ground truth tree whose set of vertices is G and assuming an equal edit cost of 1 for all operations, the Tree Error Rate (TER) is defined as

$$TER = \frac{S + D + I}{\|G\|} \quad (3)$$

where S, D and I are the number of substitution, deletion and insertion operations required to produce the ground truth tree from the predicted tree. This metric is designed mostly for benchmarking and is defined by analogy to the ubiquitous Symbol Error Rate (SER).

5.4 Tier 3: Semantic reconstruction

This tier considers whether the subset of music semantics required by MTN has been extracted correctly. It depends on the matching extracted from the structural level in order to identify the association between objects and their ground truth correspondences. Since evaluation has to be possible at the measure level, the graphical definition of pitch is used, which is the position of the notehead within the system.

A note n is defined as a 3-tuple $(p, t, d) \in \mathbb{Z}^2 \times \mathbb{Q} \times \mathbb{Q}$, where p is a tuple of the staff the note belongs to and its position within the staff, t is the number of beats from the start of the measure and d is the duration of the note in number of beats. Rests are considered notes whose pitch is irrelevant (in practical terms, they can be thought of having a unique shared number for pitch). Given the set of predicted notes P and the set of ground truth notes G, the matching between P and G is defined as a set of tuples $M = (n_p, n_g) \in P \times G$ such that at most one tuple with either a specific n_p or n_g exists.

The Missing Note Rate (MNR) is defined as the ratio of ground truth notes that do not have a corresponding prediction. More formally,

$$MNR = \frac{\|\left\{n_g \in G : (n_p, n_g) \notin M, \forall n_p \in P\right\}\|}{\|G\|}. \quad (4)$$

Similarly, the False Positive Rate (FPR) is defined as the ratio of predicted notes that do not have a corresponding ground truth note. It is defined by analogy to MNR.

Pitch Precision (PP) is defined as the number of correctly predicted pitches w.r.t. the ground truth as defined by the matching M. Therefore,

$$PP = \frac{\| \left\{ (n_p, n_g) \in M : p_{n_p} = p_{n_g} \right\} \|}{\|M\|}. \tag{5}$$

Time Precision (TP) is defined analogously.

The Average Pitch Shift (APS) is defined as the average offset in pitch from the predicted note w.r.t. its corresponding ground truth note as defined by the matching M. Therefore,

$$APS = \frac{1}{\|M\|} \sum_{\forall (n_p, n_g) \in M} p_{n_p} - p_{n_g}. \tag{6}$$

Time Average Shift (TAS) is defined analogously. Signedness is kept in order to identify the direction in which the underlying OMR system tends to move the notes.

In order for all of these metrics to be independent of the sequence length, they should be computed and accumulated for the entire dataset and not averaged on a by-prediction basis.

There is also the possibility of extending the TER metric with some semantics in order to ensure better matching and more fine-grained information from it. Instead of matching the predicted and ground truth trees with their structure alone, it would be interesting to incorporate pitch into the notehead elements. Thus, a pitch change can be though of a partial substitution of the node at a reduced cost of 0.5 for either the staff, the position on the staff or the notehead type. If more than one of this properties change, it should be considered a full substitution.

6 The COMREF dataset

We have developed a dataset built on transcriptions of public domain works as a proof of concept of the notation format. In particular, we have used the OpenScore project's transcriptions of widely known works such as The Art of the Fugue by J.S. Bach or the Planets by Gustav Holst, among others. We have also incorporated the Lieder Corpus [27] and the String Quartet corpus [50]. All these scores are engraved from MusicXML files, but we plan on supporting MEI as a conversion source at some point in the future.

6.1 Images

In summary, the dataset is produced by processing of 894 individual works into images at the measure level (including all staves that belong to it), to produce a total of 435.623 images. After verification, 461 images are removed due to noise in the annotations, totalling 435.162 images after cleanup.

The process through which the dataset was produced is summarised in Fig. 4. Scores are engraved through Verovio [42], a MEI-based score engraving system which is also compatible with MusicXML, into page-level SVG files. Using the hierarchical structure within the SVG and exploiting the optional identifier information Verovio can be instructed to attach, we employ a splitter script that finds bounding boxes for every measure and engraves them individually. It also finds what measures lie at the beginning of a line in order to properly add this information on the notation. The bounding boxes for all measures are cropped at the next staff vertically and a fixed amount of pixels horizontally.

Once the images are produced, the converter uses the MusicXML file and produces the MTN notation. In order to ensure all images have their corresponding ground truth, we use a cleaning script that finds matching identifiers for images in the MTN files. It also checks for outliers in case there are blatant mistakes in the notation. Although we have taken precautions to minimise the number of errors, there are a few images with objects far from the staff, either temporally or graphically. We remove these outliers heuristically to ensure the quality of the data.

Some qualitative examples of the generated samples can be seen in Fig. 5. The dataset purposefully crops images coarsely, incorporating parts of neighboring staves and surrounding distractors. This is in order to better emulate the result of a layout analysis algorithm, which will be inexact. The page-level images from which the measure-level dataset has been produced is also included for use cases that might require it, as well as the Verovio SVG output.

Overall, the resulting dataset is quite challenging because it incorporates a lot of variability, a selection of highly complex scores and lots of distractors.

6.2 Ground truth

The dataset has 88 classes which are represented very unevenly. Noteheads, beams, stems and flags are the most common, adding up to 70% of the entire dataset on their own. Of these, 71 are represented in a predefined test partition containing 52,884 measures.

It includes music from a wide range of sources, including scores that are monophonic, polyphonic and pianoform. Diverse instrumentations and styles are also covered, with works for large orchestras coexisting with small-scope chamber music or solo works.

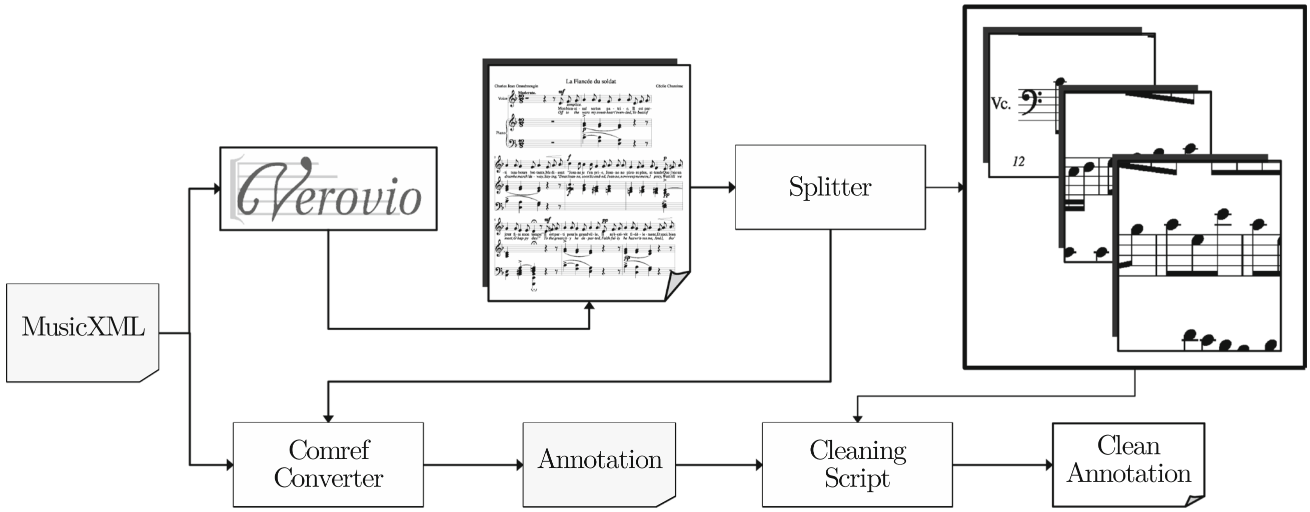

Fig. 4 The pipeline through which the COMREF dataset has been generated

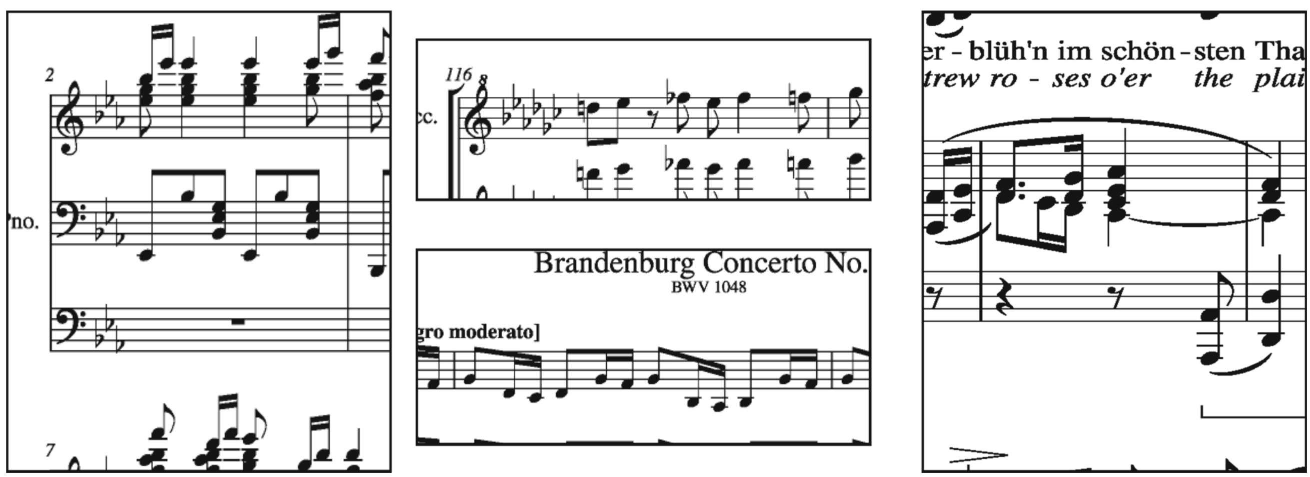

Fig. 5 Some example measures sampled from the dataset. The dataset contains scores of varying types. On the left, an example of a pianoform score with three staves. In the middle, two examples of single-staff scores where the top one is part of a system of staves. On the right, a two-staff pianoform with two staves

6.3 Baseline experiment

We conducted a simple proof-of-concept experiment on the COMREF dataset to assess the feasibility of the methodology proposed in this paper. For this purpose, we used an off-the-shelf OMR system to produce a transcription of the test partition and we analysed its results.

The OMR system used for this experiment is Audiveris [41], an Open Source page-level system capable of generating a MusicXML output from a single input image. We used this model because it can generate a full score output in MusicXML, it can be run on the full batch of images directly and offers decent results. Other systems in the community or the literature either omit the semantic reconstruction step or cannot be run on the thousands of images of the test set efficiently. Implementing the score reconstruction step for low-level intermediate music representations—e.g., bounding boxes, text representations—is outside the scope of this work. Moreover, few models are designed to address multiple-staff polyphonic music.

The page-level images of the dataset are used as input, since Audiveris requires the information of the clef, key and beat. The output MusicXML is then converted to MTN and a simple matching between predicted and ground truth samples is generated by imposing a top-down reading order given the samples known to be present on each page. If a prediction has more measures per page than the ones in the ground truth, the extra ones are just discarded.

With the setup outlined above, Audiveris predicted 45,822 measures from the 52,884 present on the ground truth. Out

Table 3 Results for Tier 1 computation on the Dataset's test partition

Token	Precision	Recall	Counts	Prop	Token	Precision	Recall	Counts	Prop
Notehead_black	0.946	0.791	193,879	0.2992	Barline_tok_heavy	0.812	0.913	643	0.0010
Stem_down	0.925	0.816	99,692	0.1539	Dyn_pp	0.901	0.714	597	0.0009
Stem_up	0.896	0.756	78,848	0.1217	Fermata	0.960	0.577	589	0.0009
Beam	0.915	0.769	48,918	0.0755	Tuplet_start	0.239	0.393	563	0.0009
Flag	0.922	0.778	28,190	0.0435	Dyn_ff	0.955	0.813	465	0.0007
Notehead_white	0.582	0.715	20,330	0.0314	Dyn_mf	0.891	0.742	430	0.0007
Accidental_sharp	0.941	0.507	19,882	0.0307	Trill	0.884	0.637	408	0.0006
Slur_start	0.852	0.685	17,329	0.0267	Rest_maxima	0.126	0.648	244	0.0004
Slur_stop	0.850	0.675	17,320	0.0267	Dyn_sfz	0.900	0.117	230	0.0004
Accidental_flat	0.907	0.412	15,350	0.0237	Wavy_line	0.000	0.000	200	0.0003
Rest_eighth	0.931	0.742	13,821	0.0213	Notehead_cue_black	0.000	0.000	185	0.0003
Dot	0.904	0.685	10,894	0.0168	Rest_32nd	0.957	0.817	164	0.0003
Staccato	0.637	0.641	9150	0.0141	Timesig_common	0.464	0.634	131	0.0002
Accidental_natural	0.926	0.738	8537	0.0132	Rest_whole	0.082	0.047	106	0.0002
Rest_quarter	0.920	0.794	8092	0.0125	Repeat_backward	0.653	0.752	105	0.0002
Tied_stop	0.822	0.496	7784	0.0120	Accidental_double_flat	1.000	0.077	104	0.0002
Tied_start	0.812	0.466	7745	0.0120	Accidental_double_sharp	1.000	0.646	96	0.0001
Clef_G	0.882	0.354	5974	0.0092	Clef_oct_F	0.000	0.000	81	0.0001
Clef_F	0.554	0.243	3120	0.0048	Dyn_fz	0.000	0.000	78	0.0001
Rest_long	0.547	0.695	3039	0.0047	Dyn_mp	0.846	0.786	70	0.0001
Wedge_stop	0.892	0.514	2849	0.0044	Clef_oct_G	0.279	0.185	65	0.0001
Rest_breve	0.861	0.519	2761	0.0043	Repeat_forward	0.295	0.316	57	0.0001
Accent	0.802	0.677	2243	0.0035	Dyn_sffz	0.000	0.000	47	0.0001
Barline_tok_regular	0.787	0.899	2023	0.0031	Dyn_ppp	0.000	0.000	44	0.0001
Dyn_p	0.771	0.716	1707	0.0026	Timesig_cut	0.818	0.878	41	0.0001
Dyn_f	0.867	0.698	1669	0.0026	Dyn_fff	0.000	0.000	39	0.0001
Arpeggiate	0.883	0.062	1573	0.0024	Dyn_rf	0.000	0.000	35	0.0001
Rest_half	0.908	0.803	1488	0.0023	Dyn_fp	0.947	0.581	31	0.0000
Wedge_diminuendo	0.874	0.641	1445	0.0022	Notehead_breve	0.000	0.000	24	0.0000
Wedge_crescendo	0.907	0.710	1433	0.0022	Notehead_cue_white	0.000	0.000	11	0.0000
Rest_16th	0.936	0.819	1418	0.0022	Segno	0.000	0.000	2	0.0000
Notehead_grace_black	0.250	0.001	1297	0.0020	Dyn_pppppp	0.000	0.000	1	0.0000
Clef_C	0.423	0.442	954	0.0015	Coda	0.000	0.000	1	0.0000
Dyn_sf	0.609	0.671	671	0.0010	Turn	0.083	1.000	1	0.0000
Tenuto	0.457	0.673	651	0.0010	Caesura	0.000	0.000	1	0.0000
					Total	0.894	0.733	647,965	1.0

of these, 40,622 measures from both sets could be matched together, corresponding to a coverage of 76.9%. The missed predictions are as a result of the engine failing to give an output on certain pages.

Given this setup, the results for Tier 1 are shown in Table 3 and the results for Tiers 2 and 3 are shown in Table 4.

From Tier 1, a few conclusions can be extracted. The first one is that Audiveris can find the structural music elements very reliably (noteheads, stems, beams and flags), all of them with recalls over 75%. This is consistent with the values seen in the missing note rate at roughly 20%, indicating that a majority of the detected noteheads is properly constructed and matched against noteheads in the ground truth. The system is also precise characterising the notes that can be matched, showing a pitch and time shift of roughly −0.09. This means that out of every 10 notes, only one is a single step below what it should.

The tree error rate is sensibly high, at 0.372. This is attributed particularly to the missing elements. Upon inspection of certain samples, the most recurrent error is the

Table 4 Results for Tiers 2 and 3

TER	Time shift	Pitch shift	Staff shift	Time prec	Pitch prec	Staff prec	FPR	MNR
0.372	– 0.096	– 0.091	0.022	0.802	0.749	0.963	0.097	0.216

From left to right, tree error rate, average time shift, average pitch shift, time precision, pitch precision, staff precision, false positive rate and missing note rate

omission of certain note groups, causing all of their attached elements to be missed.

The reason for the low scores in attribute elements (clefs, accidentals and time signature elements) is the fact that MusicXML does not encode start-of-line attribute elements because they are layout-dependent. By artificially inserting the tokens corresponding to start-of-sequence elements, we can adjust this to increase the recall but since the matching is sometimes imperfect we also insert a considerable amount of false positives.

Overall, even if the results for this specific tool on the dataset still leave room for improvement, we consider that our proposed format and metric fulfil their design purposes: unique representation of scores and evaluation. Therefore, we consider this simple trial successful.

7 Conclusions

In this paper we have argued for the implementation of a Common Optical Music Recognition Framework through the instauration of a notation format in which score reconstruction is independent from the recognition methodology. Moreover, the resulting scores can be evaluated fairly an unambiguously. Our proposed reification of this idea is the MTN format, which draws heavy inspiration from existing formats and adapts many ideas already present in the field while also being quite expressive and general. Its main idea is exploiting the expressiveness of graphs while keeping the ordering properties of sequences. Since this method builds upon some of the most widely used abstractions of the community (e.g., symbols as combinations of primitives, time from ordering, etc) it stands as a good candidate for a common endpoint for OMR as a whole. Of course, CWMN is a tremendously complex notation system which has been optimised and streamlined for hundreds of years. Nevertheless, we believe the subset of music that can be expressed in this format is large enough to be useful for the community.

In this work, we have also presented a concrete implementation of a set of metrics for OMR practitioners with the hopes of bringing together the community to speak the same language; a *lingua franca* thanks to which research can be shared and compared fairly and easily. We provide a simple baseline from which to demonstrate how the evaluation framework works. We also hope this might lower the barrier of entry and bring some new interest in the field.

The work that lies ahead now is building a corpus of music that can be employed with this format into a benchmark for CWMN recognition, both in typeset and handwritten domains. For typeset scores, building a corpus in MTN is as straightforward as compiling a comprehensive set of scores in MusicXML and engraving them. For handwritten music, the main challenge lies in developing techniques to reliably align MTN to the source material andfinding good transcriptions, which shall be the focus of our next efforts.

Supplementary information

Link to the code: https://github.com/CVC-DAG/comref-converter. Link to the dataset: https://datasets.cvc.uab.cat/comref/comref.zip.

Acknowledgements We gratefully thank the participation of Carles Badal and Jan Hajič Jr. in discussions that led to improvements on this paper. This work has been partially supported by the Spanish projects PID2021-126808OB-I00 (GRAIL) and CNS2022-135947 (DOLORES). Pau Torras is funded by the Spanish FPU Grant FPU22/00207 and developed part of this work while funded by the AGAUR Joan Oró FI grant 2023 FI-1-00324. The authors acknowledge the support of the Generalitat de Catalunya CERCA Program to CVC's general activities.

Author Contributions P.T. implemented the idea and wrote the manuscript following guidance and counseling from A.F. and S.B.. All authors reviewed the manuscript.

Funding Open Access Funding provided by Universitat Autonoma de Barcelona.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Alfaro-Contreras, M., Valero-Mas, J.J.: Exploiting the two-dimensional nature of agnostic music notation for neural optical music recognition. Appl. Sci. **11**(8), 3621 (2021). (**Publisher: MDPI**)
2. Alfaro-Contreras, M., Ríos-Vila, A., Valero-Mas, J.J., et al.: Decoupling music notation to improve end-to-end Optical Music Recognition. Pattern Recognit. Lett. **158**, 157–163 (2022). https://doi.org/10.1016/j.patrec.2022.04.032. (https://www.sciencedirect.com/science/article/pii/S0167865522001428)
3. Bach Digital. https://www.bach-digital.de/content/index.xed (2008). Accessed 01 March 2023
4. Baró, A., Badal, C., Fornés, A.: Handwritten historical music recognition by sequence-to-sequence with attention mechanism. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 205–210 (2020). https://doi.org/10.1109/ICFHR2020.2020.00046
5. Baró, A., Riba, P., Fornés, A.: Musigraph: optical music recognition through object detection and graph neural network. In: Porwal, U., Fornés, A., Shafait, F. (eds) Frontiers in Handwriting Recognition, Lecture Notes in Computer Science, pp. 171–184. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-21648-0_12
6. Baró, A., Riba, P., Fornés, A.: Towards the recognition of compound music notes in handwritten music scores. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 465–470 (2016). https://doi.org/10.1109/ICFHR.2016.0092
7. Baró, A., Riba, P., Calvo-Zaragoza, J., et al.: From optical music recognition to handwritten music recognition: a baseline. Pattern Recognit. Lett. **123**, 1–8 (2019). https://doi.org/10.1016/j.patrec.2019.02.029. (https://linkinghub.elsevier.com/retrieve/pii/S0167865518303386)
8. Beethoven-Haus Bonn. https://www.beethoven.de/en/archive/list (2023). Accessed 01 March 2023
9. Bitteur, H.: Audiveris. https://github.com/audiveris (2004)
10. Bui, H.N., Na, I.S., Kim, S.H.: Staff line removal using line adjacency graph and staff line skeleton for camera-based printed music scores. In: 22nd International Conference on Pattern Recognition, pp. 2787–2789 (2014). https://doi.org/10.1109/ICPR.2014.480
11. Byrd, D.A., Isaacson, E.J.: A music representation requirement specification for academia. Comput. Music J. **27**(4), 43–57 (2003). (https://muse.jhu.edu/pub/6/article/49604,publisher:TheMITPress)
12. Byrd, D., Simonsen, J.G.: Towards a standard testbed for optical music recognition: definitions, metrics, and page images. J. N. Music Res. **44**(3), 169–195 (2015). https://doi.org/10.1080/09298215.2015.1045424. (http://www.tandfonline.com/doi/full/10.1080/09298215.2015.1045424)
13. Calvo-Zaragoza, J., Oncina, J.: Recognition of pen-based music notation: the HOMUS dataset. In: 22nd International Conference on Pattern Recognition, pp. 3038–3043. Institute of Electrical and Electronics Engineers (IEEE) (2014). https://doi.org/10.1109/ICPR.2014.524
14. Calvo-Zaragoza, J., Rizo, D.: Camera-PrIMuS: neural end-to-end optical music recognition on realistic monophonic scores. In: 19th International Society for Music Information Retrieval Conference, Paris, France, pp. 248–255 (2018a). http://ismir2018.ircam.fr/doc/pdfs/33_Paper.pdf
15. Calvo-Zaragoza, J., Valero-Mas, J.J., Pertusa, A.: End-to-End Optical Music Recognition Using Neural Networks. In: Proceedings of the 18th International Society for Music Information Retrieval Conference, pp. 472–477. ISMIR Publication, Suzhou (2017b) https://doi.org/10.5281/zenodo.1418333, https://zenodo.org/record/1418333
16. Calvo-Zaragoza, J., Rizo, D.: End-to-end neural optical music recognition of monophonic scores. Appl. Sci. **8**(4), 606 (2018). https://doi.org/10.3390/app8040606. (https://www.mdpi.com/2076-3417/8/4/606,number:4Publisher:MultidisciplinaryDigitalPublishingInstitute)
17. Calvo-Zaragoza, J., Pertusa, A., Oncina, J.: Staff-line detection and removal using a convolutional neural network. Mach. Vis. Appl. **28**, 665–674 (2017). https://doi.org/10.1007/s00138-017-0844-4
18. Calvo-Zaragoza, J., Hajič, J., Jr., Pacha, A.: Understanding optical music recognition. ACM Comput. Surv. **53**(4), 1–35 (2021). https://doi.org/10.1145/3397499
19. Cardoso, J.S., Capela, A., Rebelo, A., et al.: Staff detection with stable paths. IEEE Trans. Pattern Anal. Mach. Intell. **31**(6), 1134–1139 (2009). https://doi.org/10.1109/TPAMI.2009.34
20. Edirisooriya, S., Dong, H.W., McAuley, J, et al.: An empirical evaluation of end-to-end polyphonic optical music recognition. arXiv:2108.01769 (2021)
21. Egozy, E., Clester, I.: Computer-assisted measure detection in a music score-following application. In: Calvo-Zaragoza, J., Pacha, A., Shatri, E. (eds) Proceedings of the 4th International Workshop on Reading Music Systems, pp. 33–36 (2022). arXiv:2211.13285, https://sites.google.com/view/worms2022/proceedings
22. Fornés, A., Dutta, A., Gordo, A., et al.: The 2012 music scores competitions: staff removal and writer identification. In: Kwon, Y.B., Ogier, J.M. (eds) Graphics Recognition. New Trends and Challenges, pp. 173–186. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-36824-0_17
23. Foscarin, F., Jacquemard, F., Fournier-S'niehotta, R.: A diff procedure for music score files. In: Proceedings of the 6th International Conference on Digital Libraries for Musicology. Association for Computing Machinery, New York, NY, USA, DLfM '19, pp. 58–64 (2019). https://doi.org/10.1145/3358664.3358671
24. Fujinaga, I.: Staff detection and removal. In: Visual Perception of Music Notation: On-line and Off Line Recognition, pp. 1–39. IGI Global (2004). https://doi.org/10.4018/978-1-59140-298-5.ch001
25. Garrido-Munoz, C., Rios-Vila, A., Calvo-Zaragoza, J.: A holistic approach for image-to-graph: application to optical music recognition. IJDAR (2022). https://doi.org/10.1007/s10032-022-00417-4
26. Good, M.: MusicXML: an internet-friendly format for sheet music. Technical reports on Recordare LLC (2001). https://pdfs.semanticscholar.org/5617/972667ff794da79a4cbb6b985e85f8487ddd.pdf
27. Gotham, M.R.H., Jonas, P.: The OpenScore lieder corpus. In: Münnich, S., Rizo, D. (eds) Music Encoding Conference Proceedings 2021, pp. 131–136. Humanities Commons (2022). https://doi.org/10.17613/1my2-dm23
28. Hajič Jr. J., Novotný, J., Pecina, P., et al.: Further steps towards a standard testbed for optical music recognition. In: Mandel, M., Devaney, J., Turnbull, D., et al (eds) 17th International Society for Music Information Retrieval Conference. New York University, New York, USA, pp. 157–163 (2016). https://wp.nyu.edu/ismir2016/event/proceedings/
29. Hajič Jr. J.: A Case for Intrinsic Evaluation of Optical Music Recognition. In: Calvo-Zaragoza, J., Hajič Jr. J., Pacha, A. (eds) 1st International Workshop on Reading Music Systems, Paris, France, pp. 15–16 (2018). https://sites.google.com/view/worms2018/proceedings
30. Hajič, J., Pecina, P.: The MUSCIMA++ dataset for handwritten optical music recognition. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 39–46 (2017). https://doi.org/10.1109/ICDAR.2017.16
31. Huang, Z., Jia, X., Guo, Y.: State-of-the-art model for music object recognition with deep learning. Appl. Sci. **9**(13), 2645–

2665 (2019). https://doi.org/10.3390/app9132645. (https://www.mdpi.com/2076-3417/9/13/2645)
32. Huron, D.: The humdrum toolkit for computational music analysis | humdrum. https://www.humdrum.org/ (2002). Accessed 14 March 2024
33. JoaoFelipe/apted: Python APTED algorithm for the tree edit distance. https://github.com/JoaoFelipe/apted/tree/master (2017). Accessed 10 March 2024
34. Mengarelli, L., Kostiuk, B., Vitório, J.G., et al.: OMR metrics and evaluation: a systematic review. Multimed. Tools Appl. (2019). https://doi.org/10.1007/s11042-019-08200-0
35. Pacha, A.: Advancing OMR as a community: best practices for reproducible research. In: Calvo-Zaragoza, J., Hajič, Jr. J., Pacha, A. (eds) 1st International Workshop on Reading Music Systems, Paris, France, pp. 19–20 (2018). https://sites.google.com/view/worms2018/proceedings
36. Pacha, A.: The challenge of reconstructing digits in music scores. In: Calvo-Zaragoza, J., Pacha, A. (eds) Proceedings of the 3rd International Workshop on Reading Music Systems, Alicante, Spain, pp. 4–7 (2021). https://sites.google.com/view/worms2021/proceedings
37. Pacha, A.: The OMR datasets project (2017). https://apacha.github.io/OMR-Datasets
38. Pacha, A., Hajič, J., Jr., Calvo-Zaragoza, J.: A baseline for general music object detection with deep learning. Appl. Sci. **8**(9), 1488–1508 (2018). https://doi.org/10.3390/app8091488
39. Parada-Cabaleiro, E., Batliner, A., Baird, A. et al.: The SEILS dataset: symbolically encoded scores in modern-early notation for computational musicology. In: 18th International Society for Music Information Retrieval Conference, Suzhou, China (2017). https://ismir2017.smcnus.org/wp-content/uploads/2017/10/14_Paper.pdf
40. Pawlik, M., Augsten, N.: Tree edit distance: robust and memory-efficient. Inf. Syst. **56**, 157–173 (2016). https://doi.org/10.1016/j.is.2015.08.004
41. Project, A.: Audiveris—open-source optical music recognition. https://github.com/Audiveris/audiveris/ (2023) (Version 5.3.1). Accessed 14 March 2024
42. Pugin, L.: Verovio, a music notation engraving library. https://www.verovio.org/ (2017). Accessed 14 March 2024
43. Rebelo, A., Fujinaga, I., Paszkiewicz, F., et al.: Optical music recognition: state-of-the-art and open issues. Int. J. Multimed. Inf. Retr. **1**(3), 173–190 (2012). https://doi.org/10.1007/s13735-012-0004-6. (http://link.springer.com/10.1007/s13735-012-0004-6)
44. Ríos-Vila, A., Iñesta, J.M., Calvo-Zaragoza, J.: End-to-end full-page optical music recognition of monophonic documents via score unfolding. In: Calvo-Zaragoza, J., Pacha, A., Shatri, E. (eds) Proceedings of the 4th International Workshop on Reading Music Systems, pp. 20–24 (2022a). arXiv:2211.13285, https://sites.google.com/view/worms2022/proceedings
45. Ríos-Vila, A., Iñesta, J.M., Calvo-Zaragoza, J.: On the use of transformers for end-to-end optical music recognition. In: Pinho, A.J., Georgieva, P., Teixeira, L.F., et al (eds) Pattern Recognition and Image Analysis, Lecture Notes in Computer Science, pp 470–481. Springer International Publishing, Cham (2022b). https://doi.org/10.1007/978-3-031-04881-4_37
46. Ríos-Vila, A., Rizo, D., Iñesta, J.M., et al.: End-to-end optical music recognition for pianoform sheet music. Int. J. Document Anal. Recognit. (IJDAR) **26**(3), 347–362 (2023). https://doi.org/10.1007/s10032-023-00432-z
47. Roland, P.: The music encoding initiative (MEI). In: 1st International Conference on Musical Applications Using XML, pp. 55–59 (2002). https://pdfs.semanticscholar.org/7fc4/16754b0508837dde8b505b3fd4dc517c7292.pdf
48. Shatri, E., Fazekas, G.: DoReMi: first glance at a universal OMR dataset. In: Calvo-Zaragoza, J., Pacha, A. (eds) Proceedings of the 3rd International Workshop on Reading Music Systems, Alicante, Spain, pp. 43–49 (2021). https://sites.google.com/view/worms2021/proceedings
49. Shishido, T., Fati, F., Tokushige, D., et al.: Listen to your favorite melodies with img2Mxml. In: Producing MusicXML from Sheet Music Image by Measure-based Multimodal Deep Learning-driven Assembly (2021). arXiv: org/abs/2106.12037 [cs]
50. String Quartet Corpus. https://github.com/OpenScore/StringQuartets (2023). Accessed 10 October 2023
51. The LilyPond Developement Team: LilyPond—essay on automated music engraving (2014). http://www.lilypond.org/
52. Torras, P., Baró, A., Kang, L., et al.: On the integration of language models into sequence to sequence architectures for handwritten music recognition. Proceedings of the 22nd International Society for Music Information Retrieval Conference, pp. 690–696. ISMIR (2021). https://doi.org/10.5281/zenodo.5624451, https://zenodo.org/record/5624451
53. Tuggener, L., Elezi, I., Schmidhuber, J., et al.: DeepScores—a dataset for segmentation, detection and classification of tiny objects. In: 24th International Conference on Pattern Recognition. ZHAW, Beijing, China (2018). https://doi.org/10.21256/zhaw-4255, arXiv:1804.00525
54. Tuggener, L., Emberger, R., Ghosh, A., et al.: Real world music object recognition. Trans. Int. Soc. Music Inf. Retri. (2023). https://doi.org/10.21256/zhaw-28644, https://digitalcollection.zhaw.ch/handle/11475/28644. Accepted 08 September 2023
55. Tuggener, L., Satyawan, Y.P., Pacha, A., et al.: The DeepScoresV2 dataset and benchmark for music object detection. In: Proceedings of the 25th International Conference on Pattern Recognition, Milan, Italy (2020). https://doi.org/10.21256/zhaw-20647
56. Wen, C., Zhu, L.: A sequence-to-sequence framework based on transformer with masked language model for optical music recognition. IEEE Access **10**, 118243–118252 (2022). https://doi.org/10.1109/ACCESS.2022.3220878
57. Yu, X., Han, Z., Gong, Y., et al.: The 1st tiny object detection challenge: methods and results. In: Bartoli A, Fusiello A (eds) Computer Vision—ECCV 2020 Workshops, Lecture Notes in Computer Science, pp. 315–323. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-68238-5_23
58. Zhang, J., Du, J., Dai, L.: Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 2245–2250. https://doi.org/10.1109/ICPR.2018.8546031 (2018)
59. Zhang, Y., Huang, Z., Zhang, Y., et al.: A detector for page-level handwritten music object recognition based on deep learning. Neural Comput. Appl. (2023). https://doi.org/10.1007/s00521-023-08216-6
60. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**(6), 1245–1262 (1989). https://doi.org/10.1137/0218082