NEURAL AUDIO-TO-SCORE MUSIC TRANSCRIPTION FOR UNCONSTRAINED POLYPHONY USING COMPACT OUTPUT REPRESENTATIONS

Víctor Arroyo *Jose J. Valero-Mas* *Jorge Calvo-Zaragoza* *Antonio Pertusa*

University Institute for Computing Research (IUII), University of Alicante, Spain

ABSTRACT

Neural Audio-to-Score (A2S) Music Transcription systems have shown promising results with pieces containing a fixed number of voices. However, they still exhibit fundamental limitations that constrain their applicability in wider scenarios. This work aims at tackling two of them: we introduce a novel output representation which addresses shortcomings related to the sequence-based A2S recognition framework and we report a first approximation to dealing with unconstrained polyphony. This is validated on a Convolutional Recurrent Neural Network (CRNN) with Connectionist Temporal Classification (CTC) A2S scheme using synthetic audio from string quartets and piano sonatas with intricate polyphonic mixtures. Our results, which improve fixed-polyphony state-of-the-art rates, may be considered a reference for future A2S works dealing with an unconstrained number of voices.

Index Terms— Audio-to-Score Transcription, Connectionist Temporal Classification, Unconstrained Polyphony

1. INTRODUCTION

Automatic Music Transcription (AMT), considered a key challenge in the Music Information Retrieval (MIR) field [1], is the research area that aims at converting acoustic recordings into some type of structured digital music notation [2]. Besides music storage and preservation, AMT represents an enabling element for computational processes such as music similarity or musicological analysis, among many others [3].

Due to the complexity of the task, AMT methods typically comprise several sequential steps [4] in which intermediate representations, such as multi-pitch estimations or piano-roll descriptions of the pieces, are produced. However, scarce research efforts have been devoted to achieving score-level codifications, especially when addressing music with unconstrained polyphony [5, 6], as it requires the inference of non-audible information as, for instance, rests or barlines.

Recent advances in Deep Learning have fostered the development of neural end-to-end AMT methods [7, 8]. Such approaches retrieve a score-like representation from an acoustic performance, process known as Audio-to-Score (A2S) transcription, in a single step. This paradigm avoids inherent issues to the aforementioned multi-stage pipeline, such as the need for feature design or error propagation between stages.

In [7] the first neural A2S method is proposed, where the result of a convolutional-based feature extraction stage was fed into a Sequence-to-Sequence architecture for modelling the sequence of music symbols. Similarly, inspired by the Automatic Speech Recognition field, a neural architecture based on DeepSpeech2 [9] is used in [10] to perform end-to-end monophonic transcription, which was later extended to address constrained polyphony with a constant number of voices [8]. In [11] the task is addressed with a convolutional stage for retrieving a multi-pitch representation coupled with a statistical-model-based rhythm quantization. In [12], a multitask transcription framework for obtaining both score-like and piano-roll representations is studied.

Nonetheless, given the immaturity within the A2S field, neural end-to-end frameworks are considered to perform adequately only under certain constrained conditions [11]. Thus, further research in ground questions such as architectural designs, encoding strategies or data acquisition, among others, are expected to equate the performance of these solutions to that of multi-stage A2S systems [13].

This work tackles some of the existing shortages which limit the application of neural end-to-end A2S schemes. In this regard, our contributions are: (i) the introduction of an alternative codification to solve issues related to the sequence-based learning framework, and (ii) proposing a first approximation to the transcription of pieces with varying polyphony degrees, namely unconstrained polyphony. On a final note, the code developed for the data gathering, preparation, model training, and experimentation is released for the sake of reproducibility and comparability in future A2S research.

2. METHODOLOGY

2.1. Learning framework

We consider a Convolutional Recurrent Neural Network (CRNN) scheme as in other neural A2S frameworks [8]. This architecture comprises a block of *convolutional* layers,

This paper is part of the project I+D+i PID2020-118447RA-I00 (MultiScore), funded by MCIN/AEI/10.13039/501100011033. Computational resources were provided by Valencian Government and FEDER funding through IDIFEDER/2020/003.

which learn the adequate set of features for the task, followed by a group of *recurrent* stages, which model the temporal dependencies of the feature-learning block, and a final fully-connected network with a *softmax* activation which retrieves the posteriogram to be decoded. To achieve an end-to-end scheme, the CRNN model is trained using the Connectionist Temporal Classification (CTC) loss [14] which allows training the network using unsegmented sequential data.

Formally, let $\mathcal{T} = \{(x_i, \mathbf{z}_i) : x_i \in \mathcal{X},\ \mathbf{z}_i \in \Sigma^*\}_{i=1}^{|\mathcal{T}|}$ be a set of train data where sample x_i drawn from space $\mathcal{X}$ of acoustic recordings is related to symbol sequence $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iN_i})$ from the set of possible sequences Σ^*, being Σ the symbol vocabulary, without further input-output alignment. Note that the use of the CTC loss function requires the inclusion of an additional "*blank*" symbol within the Σ vocabulary, i.e., $\Sigma' = \Sigma \cup \{blank\}$.

Given that the output fully-connected layer comprises $|\Sigma'|$, the model retrieves a posteriogram $p_i \in \mathbb{R}^{|\Sigma'| \times K}$ where K represents the number of frames given by the recurrent stage. The final prediction $\hat{\mathbf{z}}_i$ is obtained decoding p_i using a *greedy* approach which retrieves the most probable symbol per frame and a posterior mapping function which merges consecutive repeated symbols and removes the *blank* labels.

2.2. Score data preprocessing

As stated in [6], the choice of a symbolic format for encoding music data in A2S tasks is non-trivial. While there exist different codifications as, for instance, MusicXML, Lilypond or Kern, none of them were devised for neural end-to-end A2S transcription. Hence, issues such as encoding verbosity or representation limitations constitute factors which may constrain the performance of the model.

We consider the Kern standard [15] which suits A2S tasks due to its simple representation, straightforward manipulation, and high availability of data. This format encodes music scores in text-based symbolic codifications where voices are represented as columns, known as *spines*, and music events as rows. While in the simplest case each column contains a single note for a given time event, in more complex scenarios a variable number of notes may be present at the same event and column, or new spines can appear and disappear when dividing and merging the original ones using specific symbols, leading to undetermined polyphony situations.

Despite the relevance of neural end-to-end A2S, there exists no corpora comprising real recordings and their corresponding score-level annotations suitable for this transcription formulation. Thus, as in other works from the literature [12, 13], we resort to sound synthesis procedures for retrieving audio data out of symbolic music annotations. More precisely, these data are retrieved from the Humdrum repository [16, 17], which comprises a large collection of symbolic music pieces encoded in the Kern format. Note that these elements require a series of processes for adequately addressing the A2S transcription task, depicted in Figure 1.

Table 1. Examples of the compared representation strategies.

Notation	Example	Representation		
		Kern	**Román et al. [8]**	**Ours**
Notes	𝅗𝅥.C1♯	4.CCC#	{4.},{CCC#}	{4.CCC#}
	Open tie 𝅘𝅥𝅯C6♭	[16ccc-	{[},{16},{ccc-}	{[16ccc-}
	Close tie 𝅘𝅥𝅯C6♭	8ccc-]	{8},{ccc-},{]}	{8ccc-]}
	Continue tie 𝅗𝅥C6♭	2ccc-_	{2},{ccc-},{_}	{2ccc-_}
Rest	𝄽.	4.r	{4.},{r}	{4.r}
Barlines	Measure separation	=	{=}	{=}

Initially, the Kern music files undergo a cleansing step in which ornamental elements such as fermatas, beams, stems, slurs, elisions, editorial marks, rest positions, and grace notes are removed. Notes and rests, clef, key, time signature, articulation marks, and dynamic expression are kept.

The pieces are randomly split into fragments of 3-6 measures and synthesized with their specific timbre at a sampling rate of $22,050$Hz. A log Short Time Fourier Transform representation with log-spaced bins and log-scaled magnitude is obtained as input to the model. We consider A4 as the reference pitch with 48 bins per octave, a 2048-sample Hamming window (92.88ms), and a 512-sample hop size (23.22ms).

For the sake of comparison and reproduction in the A2S field, all required code and data is released for future research at https://github.com/vicarmar/audio2score.

2.3. Neural vocabulary representations

This work proposes a compact output representation which tackles shortcomings in previous encoding strategies in the Kern-based neural A2S transcription framework. More precisely, and inspired by the Automatic Speech Recognition in which word and subword codifications are used, we consider using the entire Kern tokens as categories of the Σ vocabulary.

Previous approaches [8] have considered disentangling Kern symbols into its pitch, duration, and other alterations as the Σ vocabulary. However, end-to-end A2S methods based on CRNN-CTC are constrained by the fact that, for output sequence $\mathbf{z}_i$, condition $K \geq 2\,|\mathbf{z}_i| + 1$ must be fulfilled, where K represents the length of posteriogram p_i. While this limitation may be circumvented with architectural readjustments, the proposed compact representation is expected to alleviate this issue as it produces shorter output sequences than the disentangled one together with higher performance rates.

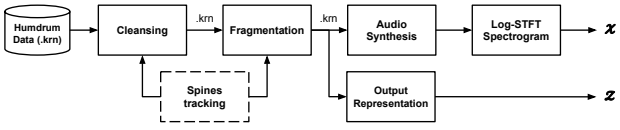

Fig. 1. Graphical of the score data preparation process.

2.4. Unconstrained polyphony

Current state-of-the-art neural A2S methods tackle varying polyphony degrees by performing a reduction process for which only the root note of a multi-voice spine or chord is kept [8]. While such approach allows modeling unconstrained polyphonic pieces as fixed-polyphony cases, it hinders the applicability of A2S in realistic scenarios. This work proposes a first approximation for explicitly dealing with variable and unconstrained polyphony.

As shown in Figure 2, variable polyphony in a Kern spine is defined by either containing multiple voices or by dividing it temporally, with specific identifiers, into several spines that can be later merged. These new paths shift all higher spines to the right in new columns.

The reduction process for constraining the polyphony degree tracks the spines base paths to locate the root note of each one, independently of how many shifts they have suffered due to previously added paths. While it adds more complexity to the fragmentation process than directly keeping the first columns according to the fixed polyphony degree required, it keeps much more musical richness from the original piece.

In the unconstrained case, the division and merge symbols are kept along with the variable number of spines. In the piece fragmentation process we ensure that the correct number of divisions and merges are present in each fragment to match the original header, regardless of where the fragment is cut. For that, the division and merge symbols prior to a fragment are tracked and added after the header of the excerpt.

Figure 2 graphically shows a score excerpt considering the existing reduction process and the proposed method.

Fig. 2. Kern fragments: On the left, Constrained polyphony, removing multiple voices and chords. On the center, Unconstrained polyphony. The original rendered score, on the right.

3. EXPERIMENTATION

3.1. Corpora

We consider different corpora for the evaluation, which are summarized in Table 2 in terms of duration and vocabulary size. The first one—*Quartets*—corresponds to that used in [8] and comprises a set of string quartets from Haydn, Mozart, and Beethoven, retrieved from the Humdrum repository [16]. It does not contain large variations in number of voices as it was considered for fixed-polyphony scenarios.

Table 2. Corpora description in terms of duration (hours) and vocabulary size depending on the codification considered.

Corpus	Duration (h.)	Vocabulary	
		Román et al. [8]	Ours
Quartets	71.68	148	9,147
Sonatas	53.54	173	13,632
Combined	125.22	173	13,632

The second set, referred to as *Sonatas*, was specifically compiled for A2S as part of this work. It comprises piano sonatas from Beethoven, Mozart, and Scarlatti retrieved from the Humdrum repository [16] and Haydn works from [17]. This corpus depicts a higher complexity regarding polyphony degrees with a wider tessitura and stylistic variations.

These corpora are merged for increasing the variability and complexity of the task. This set, which is referred to as *Combined*, exhibits a variable polyphony degree which ranges from 2 to 13 simultaneous voices, with a median of 4.

Finally, the corpus is divided into three partitions at file level, train (70%), validation (15%), and test (15%), being all excerpts of a piece in the same set to avoid possible biases.

3.2. Evaluation metrics

As in previous A2S transcription works, we consider the *Character Error Rate* (CER) and *Word Error Rate* (WER) figures of merit [12, 13]. These rates are computed as the number of elementary editing operations (insertions, deletions or substitutions), at character and word levels, respectively, required to convert prediction $\hat{\mathbf{z}}_i$ into reference $\mathbf{z}_i$, normalized by the running length of $\mathbf{z}_i$.

3.3. Neural model configuration

The CRNN scheme comprises two convolutional layers which apply 16 filters of size 3×3 with a stride of 2 in the frequency axis, avoiding pooling layers. To comply with the aforementioned CTC training restriction, the output from the convolutional block is splitted in half as a computationally less expensive method than increasing the spectrogram resolution. These features are fed into the recurrent stack represented by two Bidirectional Long Short-Term Memory cells with 1024 hidden units each followed by the fully-connected layer which retrieves the posteriogram to be decoded.

This model is trained with batch size of 16 elements, Stochastic Gradient Descent (SGD) optimization with Nesterov momentum of 0.9 and a learning rate schedule with an initial value of $3 \cdot 10^{-4}$, and an annealing figure of 0.91. We

Table 3. Results of Román et al. method [8] against the presented compact encoding, both restricting the number of voices—*CompC*—and its extension to unconstrained polyphony—*CompU*. Experiments comprise two scenarios: (i) train and test data consider the same corpus and (ii) models are directly trained with the *Combined* set. Bold figures represent the minimum error rates achieved for each corpus for the different train data scenarios.

Train Corpus	**Method**	**Constrained polyphony**						**Unconstrained polyphony**					
		Quartets		*Sonatas*		*Combined*		*Quartets*		*Sonatas*		*Combined*	
		WER	**CER**	**WER**	**CER**	**WER**	**CER**	**WER**	**CER**	**WER**	**CER**	**WER**	**CER**
Same as Test	Román et al. [8]	23.7	18.8	**31.6**	**21.4**	26.2	19.9	-	-	-	-	-	-
	CompC	**21.3**	**17.4**	33.3	23.6	**23.0**	**18.1**	-	-	-	-	-	-
	CompU	**21.3**	**17.4**	36.6	26.6	25.8	20.5	22.0	18.1	44.6	33.7	29.7	24.0
Combined set	Román et al. [8]	24.1	19.1	30.7	**21.0**	26.2	19.9	-	-	-	-	-	-
	CompC	**19.8**	**16.2**	**29.9**	21.6	**23.0**	**18.1**	-	-	-	-	-	-
	CompU	21.8	17.8	34.7	25.4	25.8	20.5	22.6	18.5	42.0	32.2	29.7	24.0

iterate for 50 epochs, keeping the weights which minimize the WER metric on the validation set.

3.4. Results

Table 3 presents the results obtained when comparing the state-of-the-art method in [8], only applicable to constrained polyphony cases, against our proposals: *CompC*, which considers the presented compact encoding with the voice reduction process, and *CompU*, which extends *CompC* to unconstrained polyphony. Experiments consider two scenarios: (i) train and test data consider the same corpus and (ii) models are directly trained with the *Combined* set.

As reported, when considering the constrained polyphony case, the proposed methods generally improve the reference work in figures between 1% and 3%. The sole exception is the *Sonatas* set, in which the base method achieves lower error rates, most likely due to the large vocabulary of the proposed method. Note that the *CompU* rarely achieves the best performance, which may be expected as the other two methods are devised for constrained polyphony cases.

In relation to the corpora considered as training data, it may be checked that the *Combined* set train scenario generally yields lower error rates than using the same as test. Most reasonably this effect is due to the large amount and considerable variability of data in the *Combined* corpus, which improves the performance of the recognition model.

The proposed *CompU* constitutes the first approximation to dealing with unconstrained polyphony in Kern-based neural A2S without a voice reduction process. These results, which provide a baseline for future research, may be considered as competitive since they do not remarkably differ from those of the constrained scenario despite the higher complexity in this case. It must be highlighted that the high error rate in *Sonatas* is due to its inherent difficulty and data scarcity.

Figure 3 provides a graphical example of a music excerpt transcription with the *CompU* method. The main errors committed are the insertion of a rest, the deletion of a barline, and a wrong note duration. Other errors as the misplacement of the notes in the right spine do not constitute a major drawback since they do not affect the score-rendering process. Finally, when quantitatively evaluated, this example provides a WER of 17.6% and a CER of 9.1%.

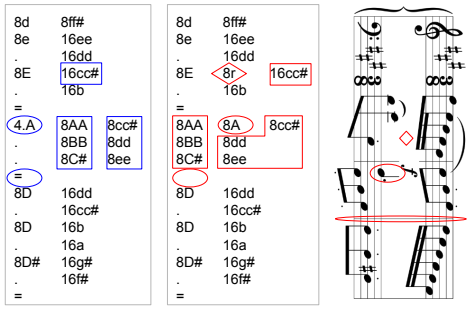

Fig. 3. Transcription example. Left and center figures show the ground truth and predicted sequences, respectively, together with the original score. Diamond, circle, and squared shapes highlight insertion, deletion or substitution, and erroneous spine assignment, respectively.

4. CONCLUSIONS

This work tackles two major limitations of neural end-to-end A2S schemes dealing with Kern encoding: on the one hand, we pose a compact output representation for addressing shortcomings related to the disentangled ones; on the other hand, we propose a first approach to deal with pieces with non-fixed polyphony degrees. The successful results obtained both prove the validity of the proposal in unconstrained polyphonic music and report a performance boost in constrained polyphony cases with respect to state-of-the-art methods. As future work, we consider extending this experimentation to real-world data, exploring Music Language Models for improving the recognition rates, and including attention-based models as an architectural alternative.

5. REFERENCES

[1] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri, "Automatic Music Transcription: Breaking the Glass Ceiling," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 379–384.

[2] Peter Grosche, Björn Schuller, Meinard Müller, and Gerhard Rigoll, "Automatic transcription of recorded music," *Acta Acustica united with Acustica*, vol. 98, no. 2, pp. 199–215, 2012.

[3] Jose J. Valero-Mas, Emmanouil Benetos, and José M. Iñesta, "A supervised classification approach for note tracking in polyphonic piano transcription," *Journal of New Music Research*, vol. 47, no. 3, pp. 249–263, 2018.

[4] Carlos Hernandez-Olivan, Ignacio Zay Pinilla, Carlos Hernandez-Lopez, and Jose R. Beltran, "A comparison of deep learning methods for timbre analysis in polyphonic automatic music transcription," *Electronics*, vol. 10, no. 7, 2021.

[5] Andrea Cogliati, David Temperley, and Zhiyao Duan, "Transcribing Human Piano Performances into Music Notation," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York, USA, 2016, pp. 758–764.

[6] Lele Liu and Emmanouil Benetos, "From audio to music notation," in *Handbook of Artificial Intelligence for Music*, Eduardo Reck Miranda, Ed., chapter 24, pp. 693–714. Springer, Switzerland, 2021.

[7] Ralf Gunter Correa Carvalho and Paris Smaragdis, "Towards End-to-end Polyphonic Music Transcription: Transforming Music Audio Directly to a Score," in *IEEE Workshop for Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2017, pp. 151–155.

[8] Miguel A. Román, Antonio Pertusa, and Jorge Calvo-Zaragoza, "A Holistic Approach to Polyphonic Music Transcription with Neural Networks," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, Netherlands, 2019, pp. 731–737.

[9] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, and Zhenyao Zhu, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," *Computer Research Repository*, vol. abs/1512.02595, 2015.

[10] Miguel A. Román, Antonio Pertusa, and Jorge Calvo-Zaragoza, "An End-to-End Framework for Audio-to-Score Music Transcription on Monophonic Excerpts," in *Proc. of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 34–41.

[11] Kentaro Shibata, Eita Nakamura, and Kazuyoshi Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Information Sciences*, vol. 566, pp. 262–280, 2021.

[12] Lele Liu, Veronica Morfi, and Emmanouil Benetos, "Joint Multi-Pitch Detection and Score Transcription for Polyphonic Piano Music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, 2021, pp. 281–285.

[13] Miguel A Román, Antonio Pertusa, and Jorge Calvo-Zaragoza, "Data representations for audio-to-score monophonic music transcription," *Expert Systems with Applications*, vol. 162, pp. 113769, 2020.

[14] Alex Graves, Santiago Fernández, Faustino Gómez, and Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[15] Craig S. Sapp, "Online database of scores in the humdrum file format," in *6th International Conference on Music Information Retrieval*, London, UK, 2005, pp. 664–665.

[16] Craig S. Sapp, "humdrum-data," `https://github.com/humdrum-tools/humdrum-data.git`.

[17] Craig S. Sapp, "haydn-keyboard-sonatas," `https://github.com/craigsapp/haydn-keyboard-sonatas.git`.