POLYPHONIC MUSIC SEQUENCE TRANSDUCTION WITH METER-CONSTRAINED LSTM NETWORKS

Adrien Ycart and Emmanouil Benetos

Centre for Digital Music, Queen Mary University of London, UK

ABSTRACT

Automatic transcription of polyphonic music remains a challenging task in the field of Music Information Retrieval. In this paper, we propose a new method to post-process the output of a multi-pitch detection model using recurrent neural networks. In particular, we compare the use of a fixed sample rate against a meter-constrained time step on a piano performance audio dataset. The metric ground truth is estimated using automatic symbolic alignment, which we make available for further study. We show that using musically-relevant time steps improves system performance despite the choice of a basic representation, although mostly because it quantises the output durations. This is an encouraging result for further investigation of musically-motivated neural network designs.

***Index Terms*—** Multi-pitch detection, automatic music transcription, music language models, long short term memory networks.

1. INTRODUCTION

Automatic music transcription (AMT) is a canonic task in Music Information Retrieval (MIR). Roughly, AMT is the task of extracting from a music recording a symbolic representation describing what notes were played and when, usually in the form of a time-pitch representation called *piano-roll*. AMT is a widely discussed topic, yet, unless it is constrained to a specific instrument and instrument model [1], it remains a challenging task, in particular in the case of polyphonic music: computers are far from carrying out this task as accurately as human experts [2].

Most AMT systems use the following workflow. First, an *acoustic model* processes the audio signal, usually via a time-frequency representation, to output a non-binary time-pitch representation, in the form of a *posteriogram*. Then, a post-processing step is applied to those estimates to obtain a binary piano-roll, typically through thresholding. While the former task has been widely discussed in the literature, the latter has received little attention until quite recently (see Section 2 for a review of existing methods).

Recurrent neural networks (RNNs) have become increasingly popular for sequence modelling in various domains such as text, speech or video [3]. In particular, Long Short-Term Memory (LSTM) [4] units' ability to, theoretically, represent dependencies between elements at arbitrarily long time-scales have made them very popular for sequence transduction, i.e. transforming a given input sequence into an output sequence. Typical examples include speech recognition, machine translation, and chord recognition.

In this paper, we propose to use a simple, single-layer LSTM network to transduct multi-pitch posteriograms into piano-rolls. Some quite complex neural architectures were developed for this purpose [5], but very often, because they do not take into account the unique nature of music signals, their potential is not fully exploited. In particular, we show that using musically-relevant time steps, such as time-steps of a sixteenth-note, instead of shorter, time-constant time steps can increase the performance of a system, although mostly because it quantises the output durations. Using tempo-related time steps requires to have at least beat annotations; in a real-life setting, those annotations would have to be obtained by a beat tracking method. In this study, we consider that the rhythmic ground truth is given (see Section 3 for details). The main contribution of this work is to show that by making musically-motivated design choices, the performance of such systems can be increased.

The paper is organised as follows. In Section 2, we review existing post-processing techniques for time-pitch representations. In Section 3, we describe the dataset we used and how it was obtained. We present the models we used in the experiments in Section 4 and the evaluation metrics in Section 5. We present the results of the experiments in Section 6. Finally, in Section 7, we discuss the results and propose some perspectives for future developments.

2. STATE OF THE ART

In the vast majority of AMT systems, a post-processing step, also known as *note tracking*, is necessary to obtain a binary piano-roll from a real-valued time-pitch representation. The most straightforward way to do so is to apply a threshold to the posteriogram, with the risk of having false alarms and missing lower-activation notes.

One of the most commonly-used post-processing techniques is described in [6]; for each note, the activation is represented as a 2-state on-off hidden Markov model (HMM). This technique is limited, as is considers each note independently, whereas pitches in music do not occur randomly: they are strongly correlated, both instantaneously and temporally. Techniques that allow to account for those complex dependencies have also been proposed. Raczyński et al. [7] designed a hierarchical model of harmony using Dynamic Bayesian Networks to post-process the output of a multi-pitch estimator. In [8], a model using linear dynamical systems was proposed to post-process multi-pitch posteriograms.

More recently, RNNs have been applied to AMT. Boulanger-Lewandowski et al. proposed in [9] an architecture for symbolic music modelling combining a Restricted Boltzmann Machine (RBM) to model instantaneous dependencies and an RNN to link them through time. This model, first proposed as a symbolic music model, was then used for AMT in [5], taking as input a Deep Belief Network-based representation of the audio signal. The same architecture was also used in [10], where it was combined with a variety of neural acoustic models. Those networks have a large number of parameters, require large amounts of training data, and can be prone to overfitting. Although good results are achieved, their design choices

A. Ycart is supported by a QMUL EECS Research Studentship. E. Benetos is supported by a UK RAEng Research Fellowship (RF/128).

are often questionable musically. In particular, the original RNN-RBM was used on quantised data, where durations are expressed in fractions of a beat. However, both in [5] and [10], the architecture was used as is, but with time steps of the order of 10ms, which is short compared to the typical duration of a musical note, and does not take into account the tempo of the music piece.

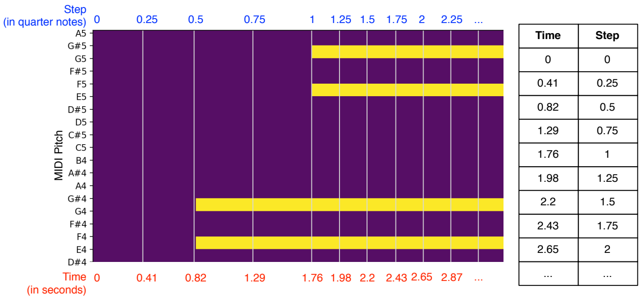
A variety of studies have looked into how, when used inappropriately, neural networks bring little improvement over simpler methods. In [11], it was shown that better results than [10] can be obtained on AMT with neural acoustic models without resorting to the RNN-RBM, simply by carefully tuning hyperparameters and using appropriate input representations. In [12], an RNN and HMM were compared on a harmony modelling task. When the frame rate is high (order of 10 fps), the RNN only has a smoothing effect, and is no more efficient than simpler temporal models such as HMMs. They suggest though that on the chord-level (i.e. one symbol per chord, no matter how long), RNNs significantly outperform HMMs. In [13], similar findings are reported for polyphonic symbolic music prediction: using a small time-step only results in a smoothing effect due to the predominance of self-transitions, and using a musically-relevant time step allows the network to learn interesting musical properties such as meter and tonality to some extent. We aim at following this direction, extending those results to AMT.

3. DATASET

For experiments on transduction, we use the MAPS dataset [14], which contains MIDI files of polyphonic piano music, along with aligned audio renditions, generated using synthetic pianos and Disklavier acoustic pianos. It contains 238 pieces of classical music (18h total duration) with some pieces performed more than once, on different pianos. Rhythmic ground truth is not available in this dataset, which is however needed for our experiments, since we aim to use a time-step of a sixteenth note.

The MAPS MIDI files were taken from from the Piano-Midi.de[1] database. This database was made by manually editing the velocities and the tempo curves of quantised MIDI files in order to give them a natural interpretation and feeling. The MIDI performances contain expressive timing, and at the same time, the rhythmic ground truth is readily available (it was however not kept in the MAPS MIDI files). We exploit this specificity to get the rhythmic ground truth from the Piano-Midi.de dataset, and use it on the MAPS MIDI files.

To do so, directly copying the rhythm information from Piano-Midi.de was not possible, since this dataset is continuously updated by its creator, meaning many files have been slightly modified since the creation of the MAPS database. We thus resort to a symbolic MIDI-to-MIDI alignment method [15] to align pairs of files. In the cases where the two versions are too different and the alignment fails, some manual editing (e.g. modifying the pitches) is made on the Piano-Midi.de files to make them match the MAPS files, while still preserving the rhythm. Around 10 pieces were manually edited. From these alignments, we deduce for each MAPS file a table linking each sixteenth-note step to its time of occurrence, in the form of a table T such that $T[i] = [t, s]$ where t is a time in seconds and s is a sixteenth-note step. We make the set of these tables available for further use[2]. Some MAPS data was lost in the process, overall about 14 minutes of data. An example piano-roll and associated time-step correspondence table is given in Fig. 1.



Time	Step
0	0
0.41	0.25
0.82	0.5
1.29	0.75
1.76	1
1.98	1.25
2.2	1.5
2.43	1.75
2.65	2
...	...

Fig. 1. A piano-roll, along with its rhythm table. A tempo change can be noted at $t = 1.76$ (sixteenth notes get shorter).

4. MODEL

In this section, we describe the multi-pitch detection and transduction models. We use the transduction model in two configurations: using time steps of 10ms (that we call *time-based steps*), and using time steps of a sixteenth note (that we call *note-based steps*).

4.1. Acoustic Model

To obtain the posteriograms, we use the multi-pitch detection system of [16], which is based on Probabilistic Latent Component Analysis. The system decomposes an input spectrogram into several probability distributions for pitch activations, instrument source contributions and tuning deviations, using a fixed dictionary of pre-extracted spectral templates. For this experiment, a piano-specific system is used, trained using isolated notes from the MAPS database [14]. The output of the acoustic model is a real-valued matrix M of size $88 \times T$, each of the 88 rows corresponding to activations of one of the 88 keys of a piano over time, with a time step of 10ms.

In the case of note-based time steps, we have to downsample these posteriograms, in order to get one value per sixteenth note step (we remind the reader that in this study, we consider the locations of the sixteenth note marks given). Formally, we have to transform $M[p, t]$ into $N[p, s]$, where t is a time index and s is a sixteenth-note step index, given a correspondence table T. To do so, we use 3 different methods, to be described as follows. For each $T[i] = [t_1, s_1], T[i + 1] = [t_2, s_2]$ and for each pitch p :

$$\textit{avg: } N[p, s] = \frac{\sum_{n=t_1}^{t_2} M[p,n]}{t_2 - t_1}$$

$$\textit{step: } k = t_1 + \frac{t_2 - t_1}{4}, N[p, s] = \frac{\sum_{n=t_1}^{k} M[p,n]}{k - t_1}$$

$$\textit{exp: } w[n] = 0.1^{n * \frac{1}{t_2 - t_1}}, N[p, s] = \frac{\sum_{n=t_1}^{t_2} w[n] * M[p,n]}{t_2 - t_1}$$

The *step* downsampling allows to focus on the note attacks, while *exp* accounts for the exponentially-decaying nature of piano notes. The parameters of *exp* were determined heuristically.

4.2. Transduction Model

The goal of this study is to demonstrate how using musically-relevant time steps can improve the performance of a multi-pitch detection system. For simplicity, and to limit interference with other techniques, we deliberately use a simple LSTM architecture for the transduction model. In particular, we choose not to use multiple layers, nor to use dropout or any other regularisation method during training. These will be investigated in future work.

[1] http://piano-midi.de/

[2] http://c4dm.eecs.qmul.ac.uk/ycart/icassp18.html

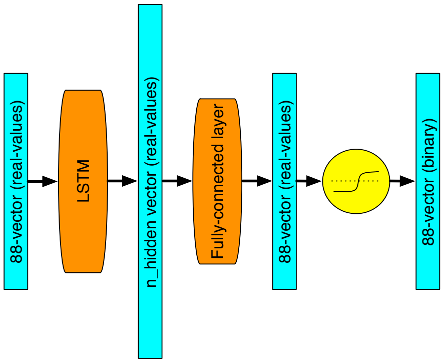


Fig. 2. Single-layer LSTM network architecture.

We thus use an LSTM with 88 inputs, one single hidden layer with N hidden nodes, and 88 outputs, one for each piano key, which are sent through a sigmoid function. An LSTM unit is defined as follows (biases are omitted for simplicity):

$$\begin{aligned} f_t &= \sigma(W_f h_{t-1} + U_f x_t) & c_t &= \tanh(W h_{t-1} + U x_t) \\ w_t &= \sigma(W_w h_{t-1} + U_w x_t) & h'_t &= h'_{t-1} \circ f_t + w_t \circ c_t \\ o_t &= \sigma(W_o h_{t-1} + U_o x_t) & h_t &= o_t \circ \tanh(h'_t) \end{aligned}$$

where $\circ$ is the elementwise product, σ is the sigmoid function, f_t, w_t, o_t are the forget, write and output functions respectively (functions of $\mathbf{R}^N$), h_t and c_t are the hidden state and the candidate at time t respectively (vectors in $\mathbf{R}^N$). The network is trained using the Adam optimiser [17], using the cross-entropy between the output of the sigmoid and the ground truth as cost function, with learning rate l. We use four sets of hyper-parameters, as a simpler alternative to extensive grid search: $N \in \{128, 256\}$ and $l \in \{0.001, 0.01\}$.

The output of the network is then thresholded to obtain a binary piano-roll. The threshold is determined by choosing the one that gives the best results on the validation dataset (see Sec. 6 for more information). The network architecture is provided in Fig. 2. An example comparing the input posteriogram, the thresholded output of the LSTM and the ground truth is available in Fig. 3.

5. EVALUATION METRICS

We evaluate the performance of our system using two sets of metrics, following MIREX guidelines [18]. With *frame metrics*, the output and the ground truth are compared frame-by-frame. With *note metrics*, the system outputs a list of notes, that are compared to the ground truth note list (using the mir_eval implementation [19]). In both cases, the precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F-measure ($\mathcal{F}$) are computed for each file, and then averaged over groups of recordings.

The metrics are computed in 3 conditions: using the time-based time steps, the note-based time steps, and in a note-to-time setting. The frame metrics are computed the same way in the 3 settings, although with different frame sizes: 10ms in *time-based* and *note-to-time* setting, and a sixteenth-note in *note-based* setting. In the *note-to-time* setting, the model computations are made with note-based time steps (a sixteenth note), and then the results are converted back to time-based steps (10ms) using the correspondence table, and compared to the ground truth, as in the time-based setting. We can only compare results using the same time steps.

Regarding note metrics, in the *time-based* setting and the *note-to-time* setting, a note is correctly detected if its pitch matches the ground truth and the onset is within 50ms of the correct onset. In the *note-based setting*, since the onsets are aligned to the metric grid, both the pitch and the onset of the note have to be exact. It should be noted that when using note-based time steps, all notes are aligned to a grid of a sixteenth note, which means that notes outside of this grid (tuplets, trills, ornaments) will be misrepresented.

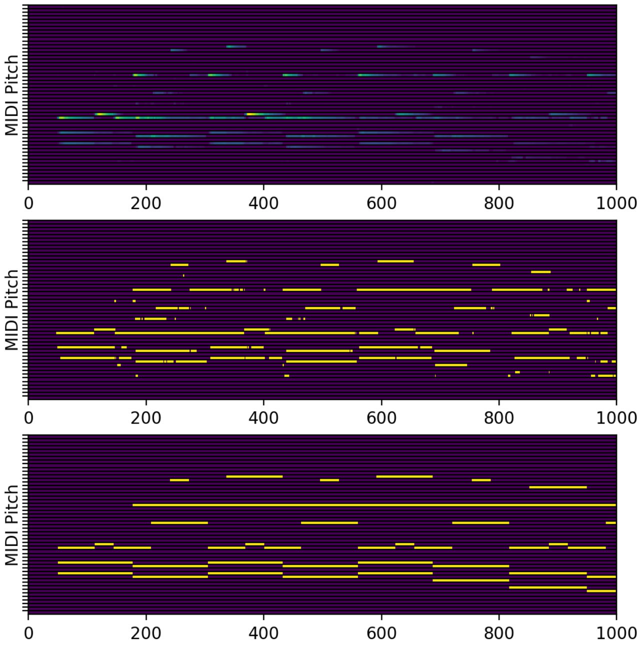


Fig. 3. An example of input posteriogram (top) and thresholded output (middle) of the LSTM, compared to the ground truth (bottom). Computations were made with 10ms time steps.

6. EXPERIMENTS

We train our transduction model using the posteriogram output of the acoustic model as input, cut in 30-second chunks. We randomly pick and set aside 15% of those chunks for validation. We train two different networks: one operating on inputs with time-based time steps, one on inputs with note-based time steps. Both networks are trained for 100 epochs. The evaluation is only performed on the 30 first seconds of each file in the test set, as is typically done in related work, to allow comparison of results. We evaluate our system using 4-fold cross-validation, using the folds referred to as *Configuration 1* in [10]. Those folds were built to have no overlap in terms of music pieces between training and testing sets, but the piano models used can be found in both sets. The acoustic model is trained using the same folds as the transduction model.

We compare our model against: median filter & thresholding (*Baseline*) and HMM smoothing [6] (*HMM*). In both above cases, model parameters were estimated on the MAPS training folds. To obtain note-based results from those systems, we downsample their binary outputs by activating a note for the considered time step if it is active for more than 5% of the corresponding sixteenth note time interval, or for more than 2 frames. We choose this criterion because of the imprecision of the alignment: sometimes, the true onset of a note occurs slightly before the time indicated in the correspondence table, which we do not want to result in a note shifted by a whole sixteenth note. The results, averaged across 4 folds, are reported in Table 1. It turns out that in the vast majority of the experiments, the

		Time-based setting			Note-based setting			Note-to-time setting		
		$\mathcal{F}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{F}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$	$\mathcal{F}(\%)$	$\mathcal{P}(\%)$	$\mathcal{R}(\%)$
Frame metrics	Baseline	63.8	71.0	61.6	69.4	70.5	71.3	65.2	64.8	69.9
	HMM	55.2	**74.1**	48.1	59.5	**76.5**	52.4	56.3	**70.5**	51.4
	LSTM	**66.3**	67.0	**67.8**	**70.2**	70.8	**71.8**	**67.1**	65.9	**71.0**
Note metrics	Baseline	**65.3**	63.2	**70.6**	**72.0**	69.3	**76.5**	**66.3**	66.6	**67.7**
	HMM	61.8	**86.2**	50.9	64.9	**85.9**	54.9	58.5	**81.9**	48.0
	LSTM	57.2	51.1	69.3	65.8	60.5	73.9	62.2	59.6	67.0

Table 1. Multi-pitch detection results for the MAPS dataset across 3 step conditions and post-processing methods.

best performing configuration for the LSTM overall was 128 hidden nodes with a learning rate of 0.01, and the best downsampling method was *step*: we only report the results for these parameters.

When evaluated on a frame basis, the LSTM is the best performing architecture. The improvement is particularly significant with time-based time steps. The LSTM gets better results in note-to-time setting compared to time-based setting, which corroborates the idea that using a musically-relevant time step improves performance.

On the other hand, with note metrics, the baseline model outperforms our system. Upon inspection of the output of the LSTM, it appears that several notes are fragmented by the system (see Fig. 3), hence the low precision. In note-based and note-to-time settings, the coarser granularity diminishes the risk of fragmentation, which improves the results, and in particular the precision. Recall slightly decreases in note-to-time setting compared to the time-based one. This is due to the fact that the size of the note-based time steps does not allow to represent non-metrical notes properly. An analysis of the dataset has determined that around 13% of the note onsets in the dataset, when quantised to a twentieth of a quarter note, do not fall on the metric grid used. The decrease in recall is thus quite low compared to the proportion of notes that could theoretically be missed.

Surprisingly, the HMM model yields quite poor results, lower than the baseline. It has the highest precision of all systems, but its recall is particularly poor. The HMM threshold was optimised on the note-based F-measure. Upon inspection of the outputs for various activation thresholds, it appears that when lowering the threshold, the HMM has a tendency to merge consecutive notes. To prevent that effect, a high threshold had to be chosen, which explains the high precision and low recall. Results would have been different if the optimisation had been performed on the frame metrics.

7. DISCUSSION

In this paper, we have presented a LSTM-based system to transduct time-pitch posteriograms into piano-rolls, as a post processing step for AMT systems. We studied the influence of time-based (fixed length of 10ms) and note-based (musical length of a sixteenth note) time steps, evaluated on an updated version of the MAPS dataset [14], which now includes metrical ground truth. The metrical ground truth for MAPS was determined using a symbolic alignment method, and we make it available for further study. We compared our approach to a baseline model and an HMM-based model. Our approach outperformed both when evaluated on a frame basis, but was outperformed by the baseline approach when evaluated on a note basis. However, using note-based time steps improved the results over time-based time steps, which is an encouraging result towards using more musically-motivated system designs.

The improvement brought by the note-based time steps is twofold. It allows to better take into account dependencies between successive notes, and it quantises the transcription. To determine the relative importance of both effects, we perform another experiment. We compare the results of the note-to-time setting with a new setting, where the computations are made using the time-based LSTM, and the binary outputs are quantised using a 16th note grid as a post-processing step (majority voting over the binary time frames). The results are equivalent in terms of F-measure for both settings for frame metrics, and even slightly better for the second setting with the note metrics. This suggests that the only improvement brought by the note-based time steps is the quantisation of the output; it actually doesn't help modelling temporal dependencies, or at least, not in the current experiment. Using a more sophisticated architecture and data augmentation techniques might help the network make sense out of temporal dependencies and improve the results over a simple post-quantisation of the output.

Given that the same piano models are present in the training and testing datasets, it is also possible that the network learns how to correct the errors the acoustic model makes on a specific piano and does not generalise to other piano models. This question will be investigated in future work.

A limitation of the note-based time steps is their inability to represent notes with durations that are not an integer multiple of a sixteenth note. To take into account tuplets, we could use as time step the greatest common divisor between all the note values we wish to represent. For instance, using a time step of a 24th note would allow to represent triplets as well as sixteenth notes. The more note values we want to take into account, the smaller the time step, which could lead to the same problem as time-based time steps: mostly self transitions, and no learning of temporal dependencies [13]. Moreover, in our representation, we do not differentiate between note onsets and continuations. As a consequence, repeated notes are represented the same way as held notes. We assume that by using two different symbols, we could prevent over-fragmentation of notes in the output.

Additional future directions include extending the use of note-based time steps to more complex architectures, such as the RNN-RBM [5]. Finally, in all our experiments, when using note-based time steps, we consider that the rhythmic ground truth is given. We made this choice to assess, as a proof-of-concept experiment, the improvement that note-based time steps can bring in an ideal case. A real-life system would obviously have to rely on an beat tracking algorithm to use those time steps. It will be the object of future work to assess if, even with the potential errors made by beat-tracking algorithms, the use of note-based time steps can still increase the performance of a system compared to time-based time steps.

8. REFERENCES

[1] S. Ewert and M. B. Sandler, "An augmented Lagrangian method for piano transcription using equal loudness thresholding and LSTM-based decoding," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2017.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT Press, 2016.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "High-dimensional Sequence Transduction," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3178–3182.

[6] Graham E. Poliner and Daniel P. W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 5, no. 1, pp. 154–162, Oct 2006.

[7] S. A. Raczyński, E. Vincent, and S. Sagayama, "Dynamic Bayesian networks for symbolic polyhonic pitch modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1830 – 1840, 2013.

[8] E. Benetos, "Polyphonic note and instrument tracking using linear dynamical systems," in *AES International Conference on Semantic Audio*, 2017.

[9] N. Boulanger-Lewandowski, P. Vincent, and Y. Bengio, "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription," *29th International Conference on Machine Learning*, pp. 1159–1166, 2012.

[10] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, May 2016.

[11] R. Kelz, M. Dorfer, F. Korzeniowski, S. Bock, A. Arzt, and G. Widmer, "On the Potential of Simple Framewise Approaches to Piano Transcription," *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pp. 475–481, 2016.

[12] F. Korzeniowski and G. Widmer, "On the Futility of Learning Complex Frame-Level Language Models for Chord Recognition," in *AES International Conference on Semantic Audio*, 2017.

[13] A. Ycart and E. Benetos, "A study on LSTM networks for polyphonic music sequence modelling," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 421–427.

[14] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[15] E. Nakamura, K. Yoshii, and H. Katayose, "Performance error detection and post-processing for fast and accurate symbolic music alignment," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[16] E. Benetos and T. Weyde, "An efficient temporally-constrained probabilistic model for multiple-instrument music transcription," in *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 701–707.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[18] M. Bay, A. F. Ehmann, and J. S. Downie, "Evaluation of Multiple-F0 Estimation and Tracking Systems," in *10th International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 315–320.

[19] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.