AN RNN-BASED MUSIC LANGUAGE MODEL FOR IMPROVING AUTOMATIC MUSIC TRANSCRIPTION

Siddharth Sigtia[†], Emmanouil Benetos[‡], Srikanth Cherla[‡], Tillman Weyde[‡], Artur S. d'Avila Garcez[‡], and Simon Dixon[†]

† Centre for Digital Music, Queen Mary University of London

‡ Department of Computer Science, City University London

† { s.s.sigtia, s.e.dixon}@qmul.ac.uk

‡ { emmanouil.benetos.1,srikanth.cherla.1,t.e.weyde,a.garcez}@city.ac.uk

ABSTRACT

In this paper, we investigate the use of Music Language Models (MLMs) for improving Automatic Music Transcription performance. The MLMs are trained on sequences of symbolic polyphonic music from the Nottingham dataset. We train Recurrent Neural Network (RNN)-based models, as they are capable of capturing complex temporal structure present in symbolic music data. Similar to the function of language models in automatic speech recognition, we use the MLMs to generate a prior probability for the occurrence of a sequence. The acoustic AMT model is based on probabilistic latent component analysis, and prior information from the MLM is incorporated into the transcription framework using Dirichlet priors. We test our hybrid models on a dataset of multiple-instrument polyphonic music and report a significant 3% improvement in terms of F-measure, when compared to using an acoustic-only model.

1. INTRODUCTION

Automatic Music Transcription (AMT) involves automatically generating a symbolic representation of an acoustic musical signal [4]. AMT is considered to be a fundamental topic in the field of music information retrieval (MIR) and has numerous applications in related fields in music technology, such as interactive music applications and computational musicology. The majority of recent transcription papers utilise and expand *spectrogram factorisation* techniques, such as non-negative matrix factorisation (NMF) [18] and its probabilistic counterpart, probabilistic latent component analysis (PLCA) [25]. Spectrogram factorisation techniques decompose an input spectrogram of the audio signal into a product of spectral templates (that typically correspond to musical notes) and component activations (that indicate whether each note is active at a given time frame). Spectrogram factorisation-based AMT systems include the work by Bertin et al. [7], who proposed a Bayesian framework for NMF, which considers each pitch as a model of Gaussian components in harmonic positions. Benetos and Dixon [3] proposed a convolutive model based on PLCA, which supports the transcription of multiple-instrument music and supports tuning changes and frequency modulations (modelled as shifts across log-frequency).

In terms of connectionist approaches for AMT, Nam et al. [20] proposed a method where features suitable for transcribing music are learned using a deep belief network consisting of stacked restricted Boltzmann machines (RBMs). The model performed classification using support vector machines and was applied to piano music. Böck and Schedl used recurrent neural networks (RNNs) with Long Short-Term Memory units for performing polyphonic piano transcription [8], with the system being particularly good at recognising note onsets.

There is no doubt that a reliable acoustic model is important for generating accurate symbolic transcriptions of a given music signal. However, since music exhibits a fair amount of structural regularity much like language, it is natural for one to think of the possibility of improving transcription accuracy using a *music language model* (MLM) in a manner akin to the use of a language model to improve the performance of a speech recognizer [21]. In [9], the predictions of a polyphonic MLM were used to this end, which was further developed in [10], where an input/output extension of the RNN-RBM was proposed that learned to map input sequences to output sequences in the context of AMT. Both in [9] and [10], evaluations were performed using synthesized MIDI data. In [22], Raczyński et al. utilise chord and key information for improving an NMF-based AMT system in a post-processing step. A major advantage of using a hybrid acoustic + language model system is that the two models can be trained independently using data from different sources. This is particularly useful since annotated audio data is scarce while it is relatively easy to find MIDI data for training robust language models.

In the present work, we integrate a MLM with an AMT system, in order to improve transcription performance. Specifically, we make use of the predictions made by a Recurrent Neural Network (RNN) and a RNN-Neural Autore-

gressive Distribution Estimator (RNN-NADE) based polyphonic MLM proposed in [9] to refine the transcriptions of a PLCA-based AMT system [2, 3]. Information from the MLM is incorporated into the PLCA-based acoustic model as a prior for pitch activations during parameter estimation. It is observed that combining the two models in this way boosts transcription accuracy by +3% on the Bach10 dataset of multiple-instrument polyphonic music [13], compared to using the acoustic AMT system only.

The outline of this paper is as follows. The PLCA-based transcription system is presented in Section 2. The RNN-based polyphonic music prediction system that is used as a music language model is described in Section 3. The combination of the two aforementioned systems is presented in Section 4. The employed dataset, evaluation metrics, and experimental results are shown in Section 5; finally, conclusions are drawn in Section 6.

2. AUTOMATIC MUSIC TRANSCRIPTION SYSTEM

For combining acoustic and music language information in an AMT context, we employ the model of [3], which supports the transcription of multiple-instrument polyphonic music and also supports pitch deviations and frequency modulations. The model of [3] is based on PLCA, which is a latent variable analysis method which has been used for decomposing spectrograms. For computational efficiency purposes, we employ the fast implementation from [2], which utilized pre-extracted note templates that are also pre-shifted across log-frequency, in order to account for frequency modulations or tuning changes. In addition, as was shown in [24], PLCA-based models can utilise priors for estimating unknown model parameters, which will be useful in this paper for informing the acoustic transcription system with symbolic information.

The transcription model takes as input a normalised log-frequency spectrogram $V_{\omega,t}$ (ω is the log-frequency index and t is the time index) and approximates it as a bivariate probability distribution $P(\omega, t)$. $P(\omega, t)$ is decomposed into a series of log-frequency spectral templates per pitch, instrument, and log-frequency shifting (which indicates deviation with respect to the ideal tuning), as well as probability distributions for pitch, instrument, and tuning.

The model is formulated as:

$$P(\omega, t) = P(t) \sum_{p,f,s} P(\omega|s, p, f) P_t(f|p) P_t(s|p) P_t(p) \tag{1}$$

where p denotes pitch, s denotes the musical instrument source, and f denotes log-frequency shifting. $P(t)$ is the energy of the log-spectrogram, which is a known quantity. $P(\omega|s, p, f)$ denotes pre-extracted log-spectral templates per pitch p and instrument s, which are also pre-shifted across log-frequency. The pre-shifting operation is made in order to account for pitch deviations, without needing to formulate a convolutive model across log-frequency. $P_t(f|p)$ is the time-varying log-frequency shifting distribution per pitch, $P_t(s|p)$ is the time-varying source contribution per pitch, and finally, $P_t(p)$ is the pitch activation, which essentially is the resulting music transcription. As a time-frequency representation in the log-frequency domain we use the constant-Q transform (CQT) with a log-spectral resolution of 60 bins/octave [23].

The unknown model parameters ($P_t(f|p)$, $P_t(s|p)$, and $P_t(p)$) can be iteratively estimated using the expectation-maximisation (EM) algorithm [12]. For the *Expectation* step, the following posterior is computed:

$$P_t(p, f, s|\omega) = \frac{P(\omega|s, p, f) P_t(f|p) P_t(s|p) P_t(p)}{\sum_{p,f,s} P(\omega|s, p, f) P_t(f|p) P_t(s|p) P_t(p)} \tag{2}$$

For the *Maximization* step (without using any priors) unknown model parameters are updated using the posterior computed from the Expectation step:

$$P_t(f|p) \propto \sum_{\omega,s} P_t(p, f, s|\omega) V_{\omega,t} \tag{3}$$

$$P_t(s|p) \propto \sum_{\omega,f} P_t(p, f, s|\omega) V_{\omega,t} \tag{4}$$

$$P_t(p) \propto \sum_{\omega,f,s} P_t(p, f, s|\omega) V_{\omega,t} \tag{5}$$

We consider the sound state templates to be fixed, so no update rule for $P(\omega|s, p, f)$ is applied. Using fixed templates, 20-30 iterations using the update rules presented in the present section are sufficient for convergence. The output of the system is a pitch activation which is scaled by the energy of the log-spectrogram:

$$P(p, t) = P(t) P_t(p) \tag{6}$$

After performing 5-sample median filtering for note smoothing, thresholding is performed on $P(p, t)$ followed by minimum note duration pruning set to 40ms in order to convert $P(p, t)$ into a binary piano-roll representation, which is the output of the transcription system, and is also used for evaluation purposes.

3. POLYPHONIC MUSIC PREDICTION SYSTEM

Taking inspiration from speech recognition, it has been shown that a good statistical model of symbolic music can help the transcription process [11]. However there are 2 main reasons for the use of MLMs in AMT not being more common.

1. Training models that capture the temporal structure and complexity of symbolic polyphonic music is not an easy task. In speech recognition, often simple language models like n-grams work extremely well. However, music has a more complex structure and simple statistical models like n-grams and HMMs fail to model these characteristics accurately even for music with simple structure [9].

2. There is no consensus on how to incorporate this prior information within the transcription system. However, recently there have been some successful attempts at using this prior information to improve the accuracy on AMT tasks [9, 10].

In this section we discuss the details of the music prediction system and the models used. In the next section we discuss how we incorporate the predictions from these models in a PLCA-based music transcription system.

3.1 Recurrent Neural Networks

A recurrent neural network (RNN) is a powerful model for time-series data which can account for long-term temporal dependencies, over multiple time-scales when trained effectively. Given a sequence of inputs $v_1, v_2, \ldots, v_T$ each in $\mathbb{R}^n$, the network computes a sequence of hidden states $\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_T$ each in $\mathbb{R}^m$, and a sequence of predictions $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T$ each in $\mathbb{R}^k$ by iterating the equations

$$\hat{h}_t = e(W_{\hat{h}x} v_t + W_{\hat{h}\hat{h}} \hat{h}_{t-1} + b_{\hat{h}}) \quad (7)$$

$$\hat{y}_t = g(W_{y\hat{h}}) \quad (8)$$

where $W_{y\hat{h}}$, $W_{\hat{h}x}$, $W_{\hat{h}\hat{h}}$ are the weight matrices, $b_{\hat{h}}$ is the bias and e and g are activation functions which are typically non-linear and applied element-wise.

An RNN can be trained using the gradient-based Back-Propagation Through Time algorithm [27] using the exactly computable error gradients in the network. However, 1^{st} order gradient methods fail to correctly train RNNs for many real-world problems. This difficulty has been associated with what is known as the *vanishing/exploding gradients* phenomenon [6], where the errors exhibit exponential decay/growth as they are back-propagated through time. years [15, 16, 19].

However, recent work in the field of neural networks and deep learning has led to several improvements in gradient based optimization methods that make training of RNNs possible. Most notably, the Hessian Free (HF) optimization algorithm has been used to train RNNs successfully on several real world datasets, including symbolic polyphonic music data [19]. Apart from second order methods like HF, several modifications to first-order gradient based methods exist that currently form the state of the art in training RNNs [5].

3.2 Recurrent Neural Network-based models

One of the drawbacks of using RNNs to predict polyphonic symbolic music is that any output of the network, $\hat{y}_i$ at time step t, is conditionally independent of $\hat{y}_j, \forall j \neq i$ given the sequence of input vectors $v_1, v_2, \ldots, v_T$. This is a severe constraint when used for modelling polyphonic music, where notes often appear in very correlated patterns within a frame. In order to overcome this limitation, models derived from RNNs have been proposed which are better at modelling high-dimensional sequences [9, 26].

The first RNN-based model that tried to model high-dimensional sequences is the Recurrent Temporal Restricted Boltzmann Machine (RTRBM) [26]. This model was extended to the more general RNN-RBM model, where the hidden states for the RBM and RNN were not constrained to be the same. For our prediction system, we make use of a variant of the RNN-RBM, called the RNN-NADE. The only difference is that the conditional distributions at each step are modelled by a Neural Autoregressive Distribution Estimator (NADE) [17] as opposed to an RBM. As discussed in the next section, to combine the predictions with the transcription system, we need individual pitch activation probabilities at each time-step. Obtaining these probabilities from an RBM is intractable as it requires summing over all possible hidden states. However the NADE is a tractable distribution estimator and we can easily obtain these probabilities from the NADE. The NADE models the probability of occurrence of a vector p as:

$$P(p) = \prod_{i=1}^{D} P(p_i | \mathbf{p}_{<\mathbf{i}}) \quad (9)$$

where $p \in \mathbb{R}^D$,p_i is the pitch activation and $\mathbf{p}_{<\mathbf{i}}$ is the vector containing all the pitch activations p_j such that $j < i$.

In our system we utilise each of the conditional probabilities $p(p_i|\mathbf{p}_{<\mathbf{i}})$ as probabilities of the pitch activations. Although the pitch activation probabilities are only conditioned on $\mathbf{p}_{<\mathbf{i}}$, we hypothesize that this will be a better model than the RNN, where the pitch activation probabilities are completely independent. Another motivation for using the NADE is that the gradients can be computed exactly, and therefore we can employ HF optimization for training the RNN-NADE.

4. COMBINING TRANSCRIPTION AND PREDICTION

In this section, we describe the process for combining the acoustic model with the music language model for deriving an improved transcription. Firstly, the input music signal is transcribed using the process described in Section 2. The resulting piano-roll representation of the transcription system is considered to be a sequence $p_1, p_2, \ldots, p_T$ that is placed as input to the MLM presented in Section 3. For the RNN-NADE, we compute the probability $P(p_i|\mathbf{p}_{<\mathbf{i}})$ for all time frames, and use that as prior information for the combined model, with the prior information denoted as $P_{MLM}(p, t)$, where $P_{MLM}(p = i, t) = P(p_i|\mathbf{p}_{<\mathbf{i}})$. For the RNN, the prediction output is directly denoted as $P_{MLM}(p, t)$, since pitch probabilities are independent.

As shown in [24], PLCA-based models use multinomial distributions; since the Dirichlet distribution is conjugate to the multinomial, a Dirichlet prior can be used to enforce structure on the pitch activation distribution $P_t(p)$. Following the procedure of [24], we define the Dirichlet hyperparameter for the pitch activation as:

$$\alpha_t(p) \propto P_t(p) P_{MLM}(p, t) \quad (10)$$

where $\alpha_t(p)$ essentially is a pitch activation probability which is filtered through a pitch indicator function computed from the symbolic prediction step (the denominator is simply for normalisation purposes).

The recording is then re-transcribed, using as additional information the prior computed from the transcription step. The modified update for the pitch activation which replaces

(5) is given by:

$$P_t(p) \propto \sum_{\omega,f,s} P_t(p,f,s|\omega)V_{\omega,t} + \kappa\alpha_t(p) \qquad (11)$$

where κ is a weight parameter expressing how much the prior should be imposed; as in [24], the weight decreases from 1 to 0 throughout the iterations. To summarize, the transcription creates a symbolic prediction, which in turn improves the subsequent re-transcription of the music signal. An overview of the complete transcription-prediction system architecture can be seen in Fig. 1.

5. EVALUATION

5.1 Dataset

For testing the transcription system, we employ the Bach10 dataset [13], which is a freely available multi-track collection of multiple-instrument polyphonic music. It consists of ten recordings of J.S. Bach chorales, performed by violin, clarinet, saxophone, and bassoon. Pitch ground truth for each instrument is also provided. Due to the tonal and homogeneous content of the dataset (single composer, single music language), it is suitable for testing the incorporation of music language models in a multiple-instrument transcription system. For training the transcription system, pre-extracted and pre-shifted spectral templates are extracted for the instruments present in the dataset, using isolated note samples from the RWC database [14].

For training the MLMs we use the Nottingham dataset [1] , a collection of 1200 music pieces in symbolic ABC format, which contain simple chord combinations and tunes. We trained the RNN and the RNN-NADE models using both Stochastic Gradient Descent (SGD) and HF to compare performance. The inputs to both the models are sequences of length 200 where each frame of the sequence is a binary vector of length 88 which covers the full piano note range. We train both the RNN and the RNN-NADE to predict the next vector given a sequence of input vectors. We train the models by minimizing the negative log-likelihood of the sequences using the cross-entropy $\sum_i t_i \log p_i + (1 - t_i)\log(1 - p_i)$ where i sums over all the dimensions of the binary vector and t_i is the pitch target.

5.2 Metrics

For evaluating the performance of the proposed system for multi-pitch detection, we employ the precision (Pre), recall (Rec), and F-measure (F) metrics, which are commonly used in transcription evaluations [1]. As in the public evaluations on multi-pitch detection carried out through the MIREX framework [1], a detected note is considered correct if its pitch is the same as the ground truth pitch and its onset is within a 50ms tolerance interval of the ground-truth onset.

[1] ifdo.ca/∼seymour/nottingham/nottingham.html

Model	Pre
RNN (SGD)	67.89%
RNN (HF)	69.61%
RNN-NADE (SGD)	68.89%
RNN-NADE (HF)	**70.61**%

Table 1. Validation results for MLMs

5.3 Results

To validate the performance of the MLMs, we calculate the prediction precision on unseen sequences of music from the Nottingham dataset of folk melodies. We utilise 694 tracks for training, 173 tracks for validation and 170 for testing [2] . For both the RNN and RNN-NADE models we sample 10 vectors from the conditional distribution at each time-step and calculate the expected precision against the ground truth. The reported precision is found by finding the mean over the predictions of every frame. Table 1 shows the results of the validation experiments. These results are of the same order as the prediction accuracies reported in [9]. We found that for both the models, HF optimization gave better precision than SGD. Training with HF was also easier as there were less hyper parameters to be tuned when compared to SGD, where learning rate needs to be updated to make sure training is effective. The RNN models had a hidden layer of size 150, while the RNN-NADE models had a hidden layer of size 100 and the NADE consisted of a hidden layer of size 150.

Multi-pitch detection experiments are performed using the proposed system, with various configurations. A first configuration only considers the transcription system from Section 2. A second configuration takes the output of the transcription system and gives it as input to the prediction system of Section 3, where the final piano-roll is the output of the prediction step. A third configuration (presented in Section 4), re-transcribes the recording, having the prediction as a prior information for estimating the pitch activations. For the prediction system, experiments were performed using both the RNN-NADE and the RNN.

Results using the various system configurations are displayed in Table 2. It can be seen that the best performance is achieved by the 3rd configuration when using the NADE-HF model for prediction, which surpasses the acoustic-only transcription system by more than 3%. In general, it can be seen that using the prediction system as a post-processing step (2nd configuration) always leads to an improvement over the acoustic-only model (1st configuration). A similar trend can be observed when integrating the prediction information as a prior in the transcription system (configuration 3) compared to just using the prediction system as post-processing (configuration 2); an improvement is always reported. Another observation can be made when comparing the RNN-NADE with the RNN, with the former providing a clear improvement. For comparative purposes, we also trained MLMs using 500 MIDI files of J.S. Bach chorales [3] and tested the models on the

[2] http://www-etud.iro.umontreal.ca/∼boulanni/icml2012

[3] http://www.jsbchorales.net/sets.shtml

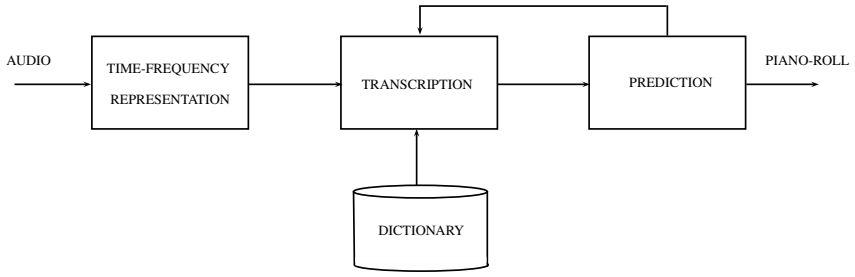

Figure 1. Proposed system diagram.

Configuration	F	Pre	Rec
Configuration 1	62.02%	58.51%	66.12%
Configuration 2 - NADE	62.62%	59.70%	65.92%
Configuration 3 - NADE	64.08%	61.96%	66.44%
Configuration 2 - RNN	62.29%	59.08%	65.98%
Configuration 3 - RNN	63.85%	61.14%	66.90%
Configuration 2 - NADE-HF	62.20%	59.14%	65.68%
Configuration 3 - NADE-HF	**65.16**%	**62.80**%	**67.78**%
Configuration 2 - RNN-HF	62.44%	59.28%	66.07%
Configuration 3 - RNN-HF	62.87%	60.03%	66.11%

Table 2. Transcription results using various system configurations.

Bach10 recordings. Using the Bach MLMs, the system reached $F = 63.58\%$, which is an improvement over the acoustic-only system, but is outperformed by the Nottingham language model.

Qualitatively, the MLMs are able to improve transcription performance by providing a rough estimate of which pitches are expected to appear in the recording (and which pitches are not expected to appear). The language models were trained using simple chord sequences (from the Nottingham dataset) that are representative of simple tonal music and are applicable as language models to the more complex Bach chorales. We believe that the reason for the J.S. Bach MLMs not performing as well as the Nottingham MLMs is due to the fact that predicting Bach's music is a complex task (many exceptions, key changes, modulations), whereas a simple tonal model like the Nottingham dataset can work as a general-purpose language model in many types of music (this is also verified in [9]).

By comparing with the method of [13] (where the Bach10 dataset was first introduced), the proposed method using the frame-based accuracy metric reaches 74.3% for the NADE-HF using the 3rd configuration, whereas the method of [13] reaches 69.7% (with unknown polyphony). As an example of the proposed system's performance, the spectrogram and raw output of the system using the 3rd configuration is displayed for a Bach10 recording Fig. 2, whereas the post-processed transcription output along with the ground truth for the same recording is shown in Fig. 3.

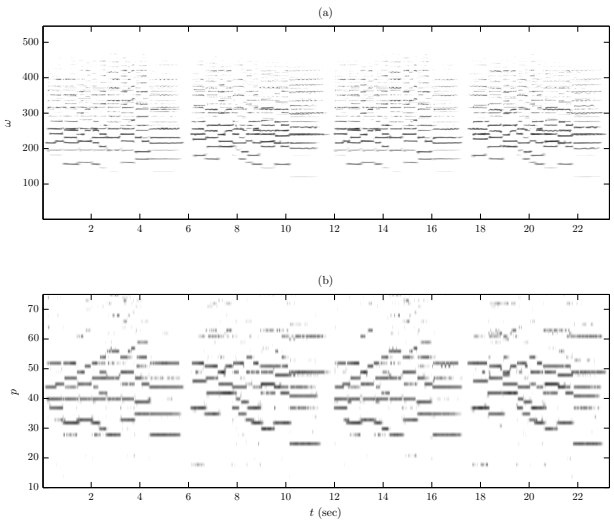

Figure 2. (a) The spectrogram $V_{\omega,t}$ for recording "Ach Lieben Christen" from the Bach10 dataset. (b) The pitch activation $P(p,t)$ using the transcription-prediction system using the 3rd configuration, with the NADE-HF.

6. CONCLUSIONS

In this paper, we proposed a system for automatic music transcription which incorporated prior information from a polyphonic music prediction model based on recurrent neural networks. The acoustic transcription model was based on probabilistic latent component analysis, and information from the prediction system was incorporated using Dirichlet priors. Experimental results using the multiple-instrument Bach10 dataset showed that there is a clear and significant improvement (3% in terms of F-measure) by combining a music language model with an acoustic model for improving the performance of the latter. These results also demonstrate that the MLM can be trained on symbolic music data from a different source as the acoustic data, thus eliminating the need to acquire collections of symbolic and corresponding acoustic data (which are scarce).

In the current system, the language models are trained on only one dataset. In the future, we would like to evaluate the proposed system using language models trained from different sources to see if this helps the MLMs generalize better. We will also investigate different system configurations, to test whether iterating the transcription and prediction steps leads to improved performance. We will also investigate the effect of using different RNN architectures like Long Short Term Memory (LSTM) and bi-directional RNNs and LSTMs. Finally, we would like to extend the current models for high-dimensional sequences to better fit the requirements for music language modelling.

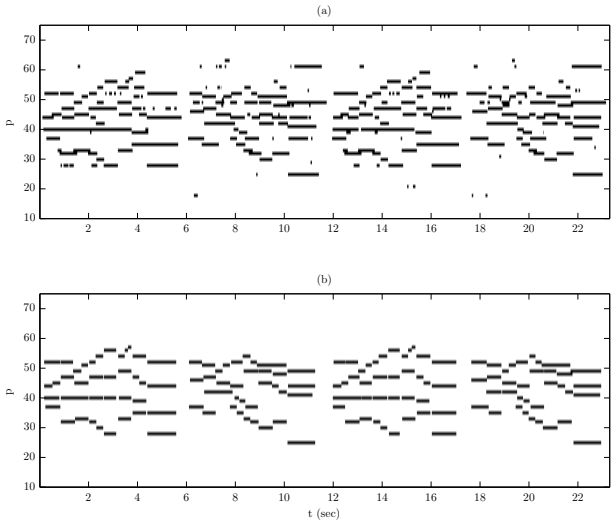

Figure 3. Transcription example for recording "Ach Lieben Christen" from the Bach10 dataset. (a) The post-processed output of the transcription-predicton system using the 3rd configuration, with the NADE-HF. (b) The pitch ground truth of the recording.

7. ACKNOWLEDGEMENT

SS is supported by a City University London Pump-Priming Grant and the Queen Mary University of London Postgraduate Research Fund. EB is supported by a City University London Research Fellowship. SC is supported by a City University London Research Studentship.

8. REFERENCES

[1] Music Information Retrieval Evaluation eXchange (MIREX). `http://music-ir.org/mirexwiki/`.

[2] E. Benetos, S. Cherla, and T. Weyde. An effcient shiftinvariant model for polyphonic music transcription. In *6th International Workshop on Machine Learning and Music*, 2013.

[3] E. Benetos and S. Dixon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, December 2013.

[5] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. Advances in optimizing recurrent networks. In *ICASSP*, pages 8624–8628, May 2013.

[6] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.

[7] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Trans. Audio, Speech, and Language Processing*, 18(3):538–549, March 2010.

[8] S. Böck and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *ICASSP*, pages 121–124, March 2012.

[9] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *29th Int. Conf. Machine Learning*, 2012.

[10] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. High-dimensional sequence transduction. In *ICASSP*, pages 3178–3182, May 2013.

[11] A. T. Cemgil. *Bayesian Music Transcription*. PhD thesis, Radboud University of Nijmegen, 2004.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[13] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Trans. Audio, Speech, and Language Processing*, 18(8):2121–2133, November 2010.

[14] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: music genre database and musical instrument sound database. In *ISMIR*, Baltimore, USA, October 2003.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] H. Jaeger. Adaptive nonlinear system identification with echo state networks. In *Advances in neural information processing systems*, pages 593–600, 2002.

[17] H. Larochelle and I. Murray. The neural autoregressive distribution estimator. *Journal of Machine Learning Research*, 15:29–37, 2011.

[18] D. D. Li and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

[19] J. Martens and I. Sutskever. Learning recurrent neural networks with Hessian-free optimization. In *28th Int. Conf. Machine Learning*, pages 1033–1040, 2011.

[20] J. Nam, J. Ngiam, H. Lee, and M. Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *ISMIR*, pages 175–180, October 2011.

[21] L. Rabiner and B.-H. Juang. Fundamentals of speech recognition. 1993.

[22] S.A. Raczynski, E. Vincent, and S. Sagayama. Dynamic Bayesian networks for symbolic polyphonic pitch modeling. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1830–1840, 2013.

[23] C. Schörkhuber and A. Klapuri. Constant-Q transform toolbox for music processing. In *7th Sound and Music Computing Conf.*, Barcelona, Spain, July 2010.

[24] P. Smaragdis and G. Mysore. Separation by "humming": user-guided sound extraction from monophonic mixtures. In *IEEE WASPAA*, pages 69–72, October 2009.

[25] P. Smaragdis, B. Raj, and M. Shashanka. A probabilistic latent variable model for acoustic modeling. In *Neural Information Processing Systems Workshop*, Whistler, Canada, December 2006.

[26] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted Boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2008.

[27] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.