Taking the Models back to Music Practice: Evaluating Generative Transcription Models built using Deep Learning

Bob L. Sturm[1] and Oded Ben-Tal[2] ⋆

[1] School of Electronic Engineering and Computer Science, Centre for Digital Music, Queen Mary University of London
[2] Department of Performing Arts, Kingston University, UK.

Abstract. We extend our evaluation of generative models of music transcriptions that were first presented in Sturm, Santos, Ben-Tal, and Korshunova (2016). We evaluate the models in five different ways: 1) at the population level, comparing statistics of 30,000 generated transcriptions with those of over 23,000 training transcriptions; 2) at the practice level, examining the ways in which specific generated transcriptions are successful as music compositions; 3) as a "nefarious tester", seeking the music knowledge limits of the models; 4) in the context of assisted music composition, using the models to create music within the conventions of the training data; and finally, 5) taking the models to real-world music practitioners. Our work attempts to demonstrate new approaches to evaluating the application of machine learning methods to modelling and making music, and the importance of taking the results back to the realm of music practice to judge their usefulness. Our datasets and software are open and available at `https://github.com/IraKorshunova/folk-rnn`.

Keywords: Deep learning, recurrent neural network (RNN), music modelling, algorithmic composition, evaluation.

1 Introduction

Deep learning (Deng & Yu, 2014; LeCun, Bengio, & Hinton, 2015) is producing much excitement in data-rich domains, e.g., image content analysis (Krizhevsky, Sutskever, & Hinton, 2012), speech processing (Hinton et al., 2012), text translation (Sutskever, Vinyals, & Le, 2014), and most recently, speech and music waveform synthesis (van den Oord et al., 2016). In our previous work (Sturm et al., 2016), as well as a variety of informal experiments,[3] we apply deep learning to model high-level textual music transcriptions within a practice termed "session", e.g., traditional dance music found in Ireland and the UK. We proposed

⋆ The authors would like to thank João Felipe Santos, Iryna Korshunova, Jeremy Keith, creator and host of thesession.org, and its many contributors, and the various musicians with whom we have worked. This work was supported in part by AHRC Grant No. AH/N504531/1.

[3] See `goo.gl/tsrSXy` and `goo.gl/PvYbov`.

two modelling approaches: *char-rnn* models transcription data one text character at a time in an endless stream; *folk-rnn* models transcription data one token at a time in a transcription-segmented fashion. Both *char-rnn* and *folk-rnn* are generative models, and so can be used to generate new transcriptions that reflect the conventions inherent to those in a training dataset. While *char-rnn* generates a stream of characters, *folk-rnn* generates individual complete transcriptions.

Sturm et al. (2016) describe our training data and methodology for creating music transcription models, but focus on evaluating *char-rnn*. In that work, we show how it can produce plausible outputs, and can be used interactively to compose music. This article complements and extends our past work by evaluating the token-based model *folk-rnn* in five different ways. As in Sturm et al. (2016), we are interested in determining or delimiting what this model has actually learned, as well as its applicability to composing music. As a first-order sanity check, we compare the descriptive statistics of our training transcriptions and transcriptions generated by *folk-rnn*. Our second approach involves analysing some of the generated transcriptions as a composition instructor would do. As a third approach, we test the generalisation limits of *folk-rnn* to particular "nefarious" initialisations, e.g., dyads, atonality, etc. Our fourth approach to evaluation applies *folk-rnn* to iterative composition in the conventions of the training data. Finally, we take the model back to music practice, within the context of music performance, and elicit feedback from a variety of practitioners.

We emphasise that *folk-rnn* is not modelling music, but instead a highly reductive abstraction removed from what one perceives as music. We thus limit our interrogation of the model to how well it understands the use of or meaning behind the elements of its vocabulary, their arrangement into larger units, and formal operations such as counting, repetition and variation. However, we are ultimately interested in the use of such models to inform and augment the human activity of music, e.g., as a component in the composition of music (Pearce, Meredith, & Wiggins, 2002). Our evaluation thus moves away from quantifying the success of a system in modelling and generating sequences of symbols, and moves toward the results of taking the application of machine learning back to the practice of music – motivated by Wagstaff (2012). We are not interested in measuring the "creativity" of the system (Loughran & O'Neill, 2016), but in 1) determining what it is actually learning to do; 2) determining how useful it is in music practice; and 3) how to make it more usable for music practice.

We organise this article in the following way. The next section augments our survey in Sturm et al. (2016) of past applications of recurrent neural networks (RNN) to music modelling and generation. We also review the approaches to evaluation in these works. Section three presents our evaluation of *folk-rnn*. We discuss these results in the fourth section. Finally, the conclusion identifies several future directions of research.

2 Previous Work in Music Modelling and Generation using Recurrent Neural Networks

Sturm et al. (2016) provide an overview of past work applying recurrent neural networks (RNNs) to music generation. We now discuss work that has appeared since then. Table 1 provides a summary of the evaluation approaches that these studies employ.

Colombo, Muscinelli, Seeholzer, Brea, and Gerstner (2016) model encoded music sequences using multilayer neural networks. They encode music as a sequence of two vectors: one encoding pitch, rest and sequence end; and the other encoding duration. They model each kind of data with two RNNs, each one having three hidden layers with 128 units. The input to the melody RNN includes the current pitch and duration of the next pitch. The input to the duration RNN is just the current duration. They train their model on a collection of 2,158 tunes (Irish, Scottish and others), all transposed to either C major or A minor, and with durations normalised over the collection. They initialise the resulting model with a portion of an existing tune, and identify similar rhythmic figures in the generated output.

Choi, Fazekas, and Sandler (2016) apply RNN with long short-term memory (LSTM) units to modelling textual representations of chord sequences, and encoded drum sequences. They take the text of 2,486 chord progressions from Realbooks and Fakebooks and train their model (two hidden layers with 512 units each) to produce one character at a time. They find that their model generates text of chord sequences that are common in Jazz. To model drum sequences, they quantise MIDI tracks to semiquaver resolution, represent each time period with a binary vector having as many dimensions as percussion instruments (nine in this case), and train an RNN to predict the drum sequence extracted from 60 Metallica songs.

A recent project of Google, *Magenta*,[4] is exploring the application of machine learning to modelling music data. A notable aspect of this project is that it is being done in a completely open and collaborative manner. So far, contributors have built a variety of models for MIDI data, including adaptations to the basic LSTM model, e.g., lookback and attention. A lookback model includes in its input explicit information about its past outputs as well as the current location within a bar. Such a model should be able to encode longer-term structures, such as repetitions of material, and resolutions of melodies. An attention model involves dynamically weighting past outputs to inform the generation of the next note.

Further work in this direction involves the application of reinforcement learning to tune RNN music models. In this case, one must define a reward function for guiding the generation of outputs that are more acceptable. Jaques, Gu, Turner, and Eck (2017) define this function by appealing to 18th-century counterpoint, e.g., melodies should begin and end on the root, a pitch should not be repeated consecutively excessively, and large leaps should resolve. They train their model

[4] `https://github.com/tensorflow/magenta`

(one hidden layer with 100 LSTM units) using homophonic melodies extracted from 30,000 MIDI songs with a semiquaver resolution, one-hot encoded in a 38-dimensional vector (36 pitches and "note off" and "no event"). They tune this generative model by using several variations of reinforcement learning, and find that the resulting models produce melodies that statistically have characteristics more favourable with regards to the compositional guidelines. Finally, they had several test subjects rate preference in a pairwise presentation of generated melodies and find that those of the tuned models are consistently preferred over the non-tuned model.

In a completely different direction, van den Oord et al. (2016) describe their work in modelling and generating audio sample data using deep convolutional networks, which do not use recurrent connections. In this case, music is not encoded, but instead embedded in the acoustic signal at 16,000 samples per second. The resulting generative model outputs one audio sample at a time. They train their model using 60 hours of piano music taken from crowd-sourced content on YouTube. The waveforms synthesised by the network exhibit realistic characteristics of piano sounds, with transients and decay.

Reference	Evaluation
Todd, 1989	Visual inspection of generated output
Mozer, 1994	1) Accuracy of system in predicting training sequences; 2) Self-auditioning of generated melodies; 3) Participants asked preference among a few melodies generated by proposed system and third-order Markov-chain
Eck & Schmidhuber, 2002	Self-auditioning of generated melodies
Chen & Miikkulainen, 2001	Basic statistics of output; self-auditioning of generated melodies
Franklin, 2006	Reproduction of training tunes, e.g., accuracy
Eck & Lapamle, 2008	Self-auditioning of generated melodies
Boulanger-Lewandowski, Bengio, & Vincent, 2012	Measurement of model log-likelihood and expected accuracy, compared with those of baseline models, for different music-sequence datasets.
Colombo et al., 2016	Visual inspection of generated output
Choi et al., 2016	Visual inspection of generated output
Jaques et al., 2017	1) Of 100,000 generated melodies, measure frequency of violation or satisfaction of specific compositional rules; 2) "Amazon Mechanical Turk in which participants were asked to rate which of two randomly selected compositions they preferred" (four melodies generated by each system)
van den Oord et al., 2016	Self-auditioning, meant only as a sanity check that the synthesis is working as expected

Table 1. Descriptions of evaluation in research applying recurrent neural networks to music modelling and generation.

3 Evaluation of the *folk-rnn* Model

We now evaluate the *folk-rnn* model, described in Sturm et al. (2016), in five different ways. We first look at the population level: we compare descriptive statistics of the training transcriptions and 30,000 generated at random by the model. We then take on the role of a composition teacher: we analyse five transcriptions randomly selected from this collection, and describe the ways in which they are and are not successful as compositions. We then act as a "nefarious tester", seeking the limits of the model's "musical knowledge" by observing its reaction to seed input that is moved more and more outside the conventions of the training data transcriptions. We then look at the model in the context of assisted music composition. Finally, we take the model to a variety of expert music practitioners to gauge its usefulness in the practice of music.

3.1 Statistical Analysis of Outputs

We now compare the descriptive statistics of our training transcriptions with those of 30,000 transcriptions we generate at random with the *folk-rnn* model. This is a straightforward way of assessing the model at a low level, but it has limited relevance to measuring how useful the model is for music practice. We thus view this perspective of evaluation as providing a first-order sanity check.

Fig. 1 compares the occurrence of specific metres, modes and number of tokens in the transcriptions. We see the model appears biased to generating transcriptions with common metre (4/4), but is biased against triple metres (6/8, 3/4, 9/8) and 2/4 and 12/8 metres. We see the model is a little biased to generating transcriptions specifying dorian mode, and less so the major mode. Finally, we see that the model is greatly biased to generating transcriptions that are 140-155 tokens long. We currently do not know what is causing these biases, but suspect that they arise from the minibatch strategy of training the model (Sturm et al., 2016).

Fig. 2 shows the distribution of pitches in transcriptions of each mode. The top part of Fig. 2 shows that for a transcription generated by the model with a major mode, we expect over 9% of the pitch tokens to be `C` (middle C), almost 26% to be `G` or `c`, and about 0.1% to be `^F`. The bottom part of Fig. 2 shows the difference between the proportions in the two populations of training and generated transcriptions. We clearly see the model has some systematic bias: it is biased toward producing pitch tokens lower in pitch than `B` (above middle C) – with a mode at `G` in mixolydian and dorian modes – and biased against higher pitches. We currently do not know the source of this bias.

Fig. 3 shows the distribution of pitch classes (scale degrees) for transcriptions denoting each mode. We can clearly see the model is producing the correct pitches in each mode, e.g., a flattened third in dorian and minor, a flattened seventh in all but major, and a flatted sixth in only minor. As expected for this kind of tonal music, the root and fifth scale degrees are the most common pitch classes. The systematic bias we see in the top part of Fig. 2 is not clear here because all pitches are folded into one octave; but we do see that the model is

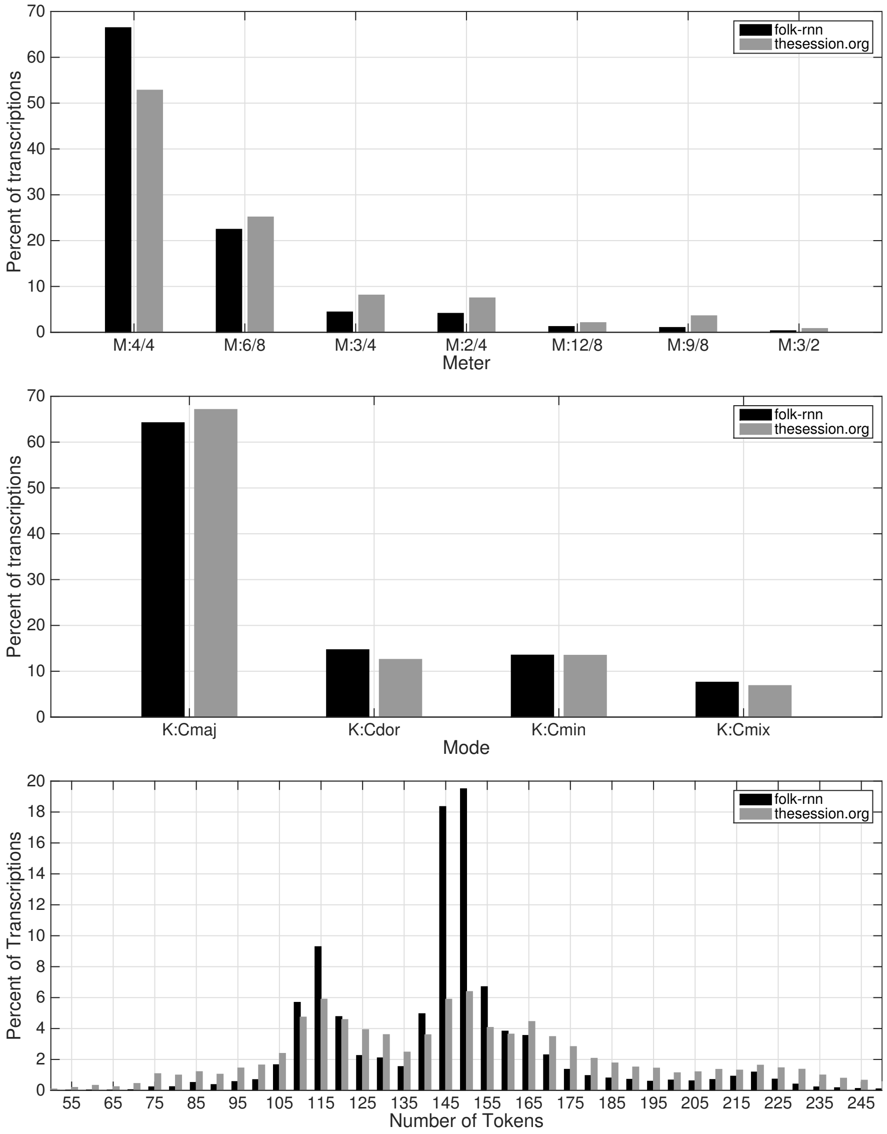

Fig. 1. Percent of transcriptions denoting each metre (top), mode (middle), and having a specific number of tokens (bottom). *Grey*: 23,635 transcriptions used for training. *Black*: 30,000 transcriptions generated by the *folk-rnn* model.

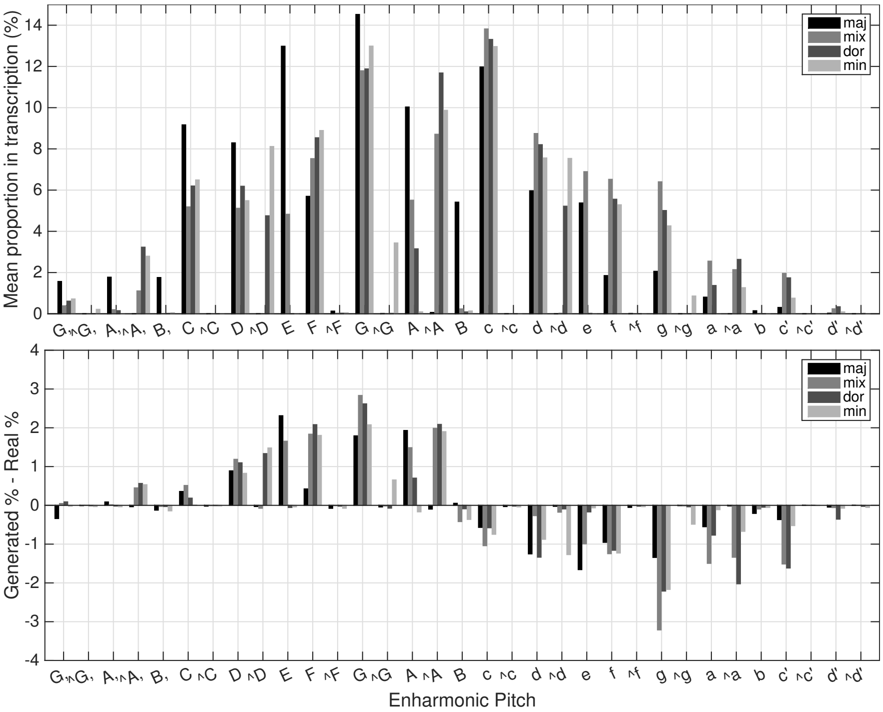

Fig. 2. Mean proportion of pitches in transcriptions of each mode, arranged chromatically from lowest pitch (left) to highest. Middle C is `C`. Top: 30,000 transcriptions generated by the *folk-rnn* model. Bottom: difference from proportions in training transcriptions. Note difference in scales.

slightly biased against the root pitch class in all modes. We currently do not know the source of this bias.

We now look at the variety of "measure token" sequences in the transcriptions as a means of assessing their forms, e.g., explicit repetition, phrase lengths, etc. We do this by extracting from each transcription its sequence of "measure tokens" (occurrences of `|, :|, |:, |1, |2`). We find 3,322 unique sequences from the training transcriptions, and only 1,867 in the generated ones. The 15 most frequent sequences in each set are shown in Tables 2 and 3. We see that the training and generated transcriptions share the top three sequences. In the case of the 23,635 training transcriptions, the top 15 sequences appear in 10,257 transcriptions; for the 30,000 generated transcriptions, they appear in 20,513. Most of these sequences show the conventional structure AABB, with each section being eight bars long, with or without pickup bars, or explicit repetition tokens at the beginning of sections. This kind of structure is common in Irish folk music (Hillhouse, 2005).

Finally, we find the following ABC errors in the 30,000 transcriptions generated by the *folk-rnn* model: 55 have the token `|1` (first ending) followed by `|1`

<table>
<tr><th>ID</th><th>Count</th><th>Measure token sequence</th></tr>
<tr><td>1</td><td>1980</td><td>|: | | | | | | | | :| |: | | | | | | | | :|</td></tr>
<tr><td>2</td><td>1739</td><td>|: | | | | | | | :| |: | | | | | | | :|</td></tr>
<tr><td>3</td><td>1109</td><td>| | | | | | | :| | | | | | | | :|</td></tr>
<tr><td>4</td><td>998</td><td>|: | | | | | | |1 :| |2 |: | | | | | | |1 :| |2 |</td></tr>
<tr><td>5</td><td>751</td><td>| | | | | | | | | | | | | | | |</td></tr>
<tr><td>6</td><td>521</td><td>| | | | | | | | | | | | | | | | |</td></tr>
<tr><td>7</td><td>441</td><td>| | | | | | | | :| | | | | | | | | :|</td></tr>
<tr><td>8</td><td>427</td><td>| | | | | | |1 :| |2 | | | | | | | |1 :| |2 |</td></tr>
<tr><td>9</td><td>416</td><td>| | | | | | | :| |: | | | | | | | :|</td></tr>
<tr><td>10</td><td>381</td><td>| | | | | | |1 :| |2 |: | | | | | | |1 :| |2 |</td></tr>
<tr><td>11</td><td>378</td><td>|: | | | :| |: | | | :|</td></tr>
<tr><td>12</td><td>303</td><td>| | | | | | | | :| |: | | | | | | | | :|</td></tr>
<tr><td>13</td><td>288</td><td>| | | :| | | | | | | | |</td></tr>
<tr><td>14</td><td>268</td><td>| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |</td></tr>
<tr><td>15</td><td>257</td><td>| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |</td></tr>
</table>

Table 2. Top 15 “measure token” sequences in training transcriptions.

<table>
<tr><th>ID</th><th>Count</th><th>Measure token sequence</th></tr>
<tr><td>1</td><td>5303</td><td>|: | | | | | | | :| |: | | | | | | | :|</td></tr>
<tr><td>2</td><td>4421</td><td>|: | | | | | | | | :| |: | | | | | | | | :|</td></tr>
<tr><td>3</td><td>2233</td><td>| | | | | | | :| | | | | | | | :|</td></tr>
<tr><td>4</td><td>1714</td><td>| | | | | | | :| |: | | | | | | | :|</td></tr>
<tr><td>5</td><td>1660</td><td>| | | | | | | | | | | | | | | |</td></tr>
<tr><td>6</td><td>1086</td><td>| | | | | | | | :| | | | | | | | | :|</td></tr>
<tr><td>7</td><td>1030</td><td>| | | | | | | | :| |: | | | | | | | | :|</td></tr>
<tr><td>8</td><td>716</td><td>| | | | | | | | | | | | | | | | |</td></tr>
<tr><td>9</td><td>586</td><td>|: | | | | | | |1 :| |2 |: | | | | | | |1 :| |2 |</td></tr>
<tr><td>10</td><td>416</td><td>|: | | | | | | | :| | | | | | | | :|</td></tr>
<tr><td>11</td><td>300</td><td>|: | | | | | | |1 :| |2 |: | | | | | | | :|</td></tr>
<tr><td>12</td><td>289</td><td>| | | | | | |1 :| |2 |: | | | | | | |1 :| |2 |</td></tr>
<tr><td>13</td><td>281</td><td>| | | | | | | | | | | | | | | | | |</td></tr>
<tr><td>14</td><td>270</td><td>|: | | | | | | | :| |: | | | | | | |1 :| |2 |</td></tr>
<tr><td>15</td><td>208</td><td>|: | | | :| |: | | | :|</td></tr>
</table>

Table 3. Top 15 “measure token” sequences in transcriptions generated by the *folk-rnn* model.

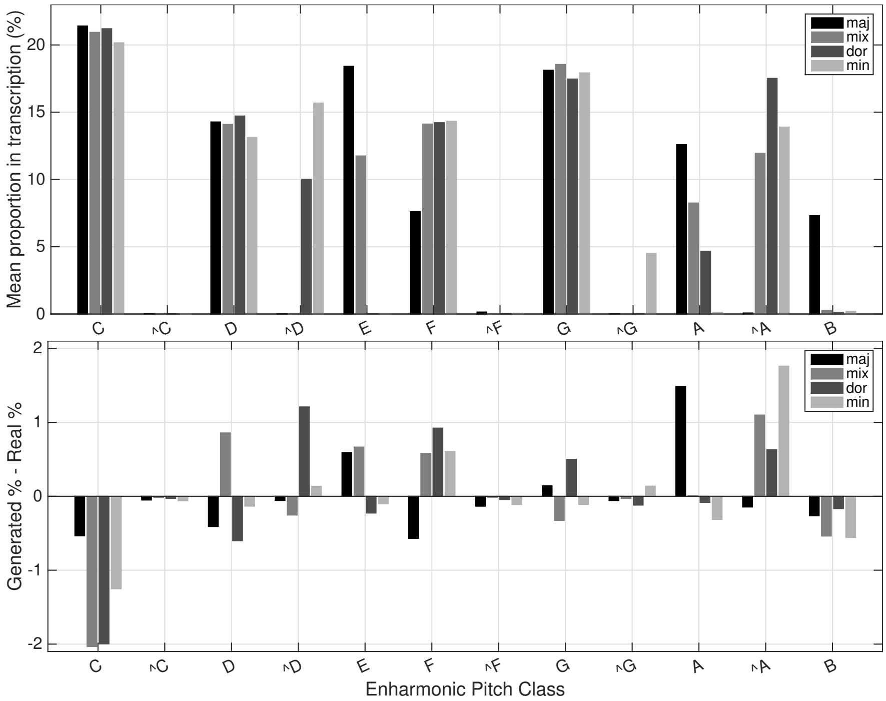

Fig. 3. Mean proportion of pitch classes in transcriptions of each mode, arranged chromatically from lowest pitch class (left) to highest. *Top*: 30,000 transcriptions generated by the *folk-rnn* model. *Bottom*: difference from proportions in training transcriptions. Note difference in scales.

instead of `|2`; 32 have only `|1` specified (without another ending), and 6 have only `|2` specified; and 17 transcriptions have incompletely specified chords, i.e., `]` appears without an accompanying `[`. We corrected such problems in the training transcriptions when creating the training data for this model (Sturm et al., 2016).

3.2 Musical Analysis of Outputs

In Sturm et al. (2016), we analyse a specific music output of the *char-rnn* model, “The Mal’s Copporim”.[5] We now analyse five transcriptions that we select randomly from the 30,000 generated by the *folk-rnn* model. (We performed no curation to select these transcriptions.) The approach we take in each analysis is to think of the model as a composition student arriving at their weekly composition tutorial with a tune. Notions of style inform our discussion about a tune, but our primary objective is to uncover what works well and what does not work in a tune.

[5] Performed by Sturm here: `https://www.youtube.com/watch?v=YMbWwU2JdLg`.

Transcription #22277 generated by the *folk-rnn* model[6] is

```
<s> M:4/4 K:Cmaj |: C B, C D E 2 F 2 | G F G A B 2 G 2 | A B c A B c d 2 |
c e d c B A G F | E 2 C D E F G 2 | A B c B A G F 2 | G A B c d 2 ^c 2 |
G 2 (3 E D E F 4 :| |: G 2 (3 c B A G A B 2 | c 2 c d e f e 2 |
d f d =B c B A G | A G F E D 2 E 2 | G 2 c B A G A B | c 2 (3 c B c
d 2 e 2 | d e c A B A G A | B A G F E 2 D 2 :| </s>
```

where we have added line breaks for readability. Fig. 4 shows the tune, which can be heard here: `http://goo.gl/qgv6xw`. We see all bars are correctly counted, but the natural in b. 11 is unnecessary. This tune has two repeated eight-bar phrases, giving it an AABB form. The opening four bars are effective: b. 1 presents an idea, b. 2 shifts it to the dominant, b. 3 continues an upward motion but breaks the rhythmic repetition with a hemiola,[7] and b. 4 brings the melody down in a stepwise fashion into the second half of A. The B phrase is not as successful as the first. It lacks musical focus and has little more than scale-wise motion. Furthermore, it lacks a clear relationship to ideas in the A phrase. The piece is clearly in C-major, but has two poor cadences in bb. 7–8 and 15–16. The c-sharp in b. 7 is entirely unexpected, and does not serve any useful function.

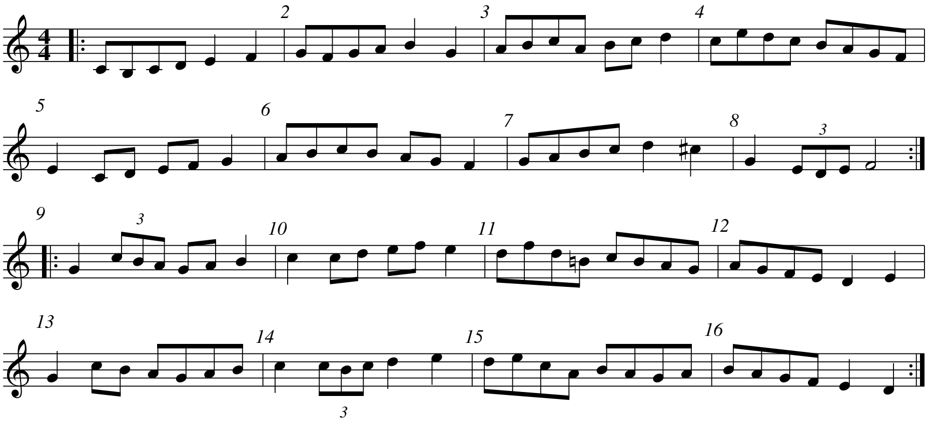

Fig. 4. Notation of transcription #22277, which can be heard here: goo.gl/qgv6xw.

Fig. 5 shows transcription #1692, which can be heard here: `http://goo.gl/5rMRLV`. We see b. 11 is missing a bar line, but all others are correctly counted, including the crotchet pickup. This piece has an AABB form with each phrase having 8 bars. The melody of each section is clearly modal (aeolian), and works fairly well, ending at the root. The highest and lowest notes appear in the B section, and are the only a-flats

[6] Unlike the *char-rnn* model, the *folk-rnn* model does not generate titles for its works. We focused its training only on key, metre, pitch, grouping, duration and "measure tokens".

[7] The hemiola here is the two groups of three ascending pitches.

in the melody. This larger span brings some focus to the overall shape of the piece. Unlike transcription #22277, the two sections here sound related, e.g., bb. 1, 5, 9 and 12 are variations. A minor problem is the somewhat awkward moment in the pickup to B and its first bar. This can be improved by moving the pitch f up a whole step to g, and moving the c in the next bar up a perfect fourth to f. Doing so moves away from middle c and supports the move to a new phrase to link better to b. 10, before descending down in the 2nd half of this phrase. A problem with this piece is that it is mostly the repetition or variation of a small pattern, and so lacks a memorable core idea. This results in a rather meandering melody.

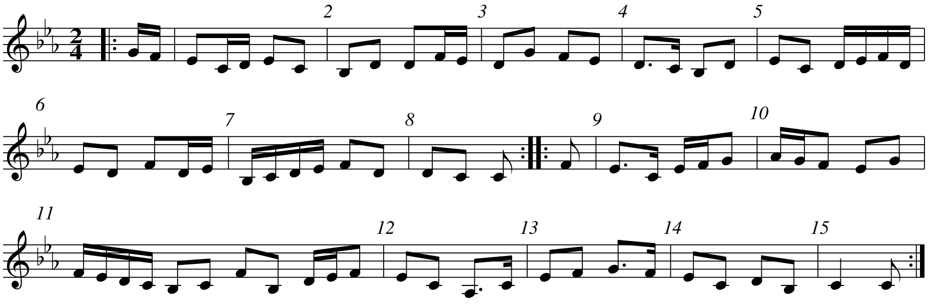

Fig. 5. Notation of transcription #1692, which can be heard here: goo.gl/5rMRLV.

Fig. 6 shows transcription #17872, which can be heard at `http://goo.gl/ChEusC`. Like the previous two tunes, this one has an AABB form, with each section consisting of eight bars. The A section ends with a satisfying V–I cadence, but the B section lacks resolution. Like the previous tune, this one repeats relatively few melodic patterns. The piece plays throughout with the rhythmic pattern in b. 1, which contributes to a coherence between the two sections, but the only four-bar unit that really works well is the first four bars of B (bb. 9–12), which are well-formed with b. 10 being a variation on b. 9. The last bar is poor, and can be improved by replacing it with b. 8. Overall, this tune sounds like an exercise for an elementary instrument lesson (though perhaps the range is too ambitious). The large melodic leap in b. 8 could be hard to play on some instruments, e.g., voice, which could be easily solved by transposing the d up one octave.

Fig. 7 shows transcription #3175, which can be heard at `https://goo.gl/7Fh5i4`. Unlike the previous three tunes, this one has an AB form, with each section having eight bars. While the previous tunes did not use enough repetition and variation to achieve coherence, this melody contains a large amount of repetition. We see similarity in the patterns appearing in bb. 2, 4 and 6. The opening two bars are varied in bb. 5–6. These, together with more-or-less workable cadences in b. 4 (to C) and b. 8 (to G), construct a plausible eight-bar phrase. Functioning almost as an answer to the first phrase, section B opens immediately in the next higher register. This leap of an eleventh is perhaps too large, and might work better if the phrase moves to the higher register more gradually. Section B also features somewhat new melodic material with the repetition of the rhythmic pattern in b. 9. Bar 13 presents a nice variation of b. 9 that has contrary motion, but b. 14 as a near repetition of b. 9 gives a sense of imbalance

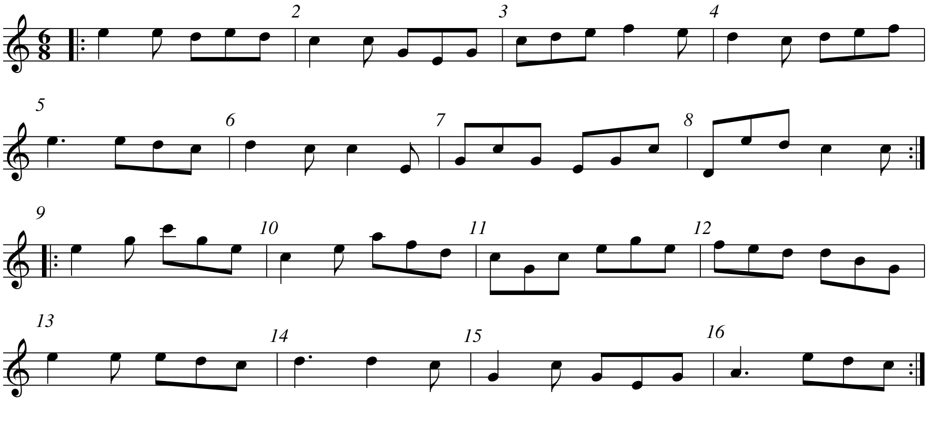

Fig. 6. Notation of transcription #17872, which can be heard here: goo.gl/ChEusC.

in the phrase that is not resolved in the final two bars; i.e., its subphrases are five and three bars long, and not a balanced four plus four. Furthermore, the segmentation suggested by the occasional crotchet breaking the running quavers creates odd length sub-phrases that are not really aligned with the 3/4 metre or the implied harmony. Thus, b. 5 becomes a more convincing pause than b. 4. Neither section cadences on the tonic, but instead skips over resolution. The A section at least slows down and lands on the dominant.

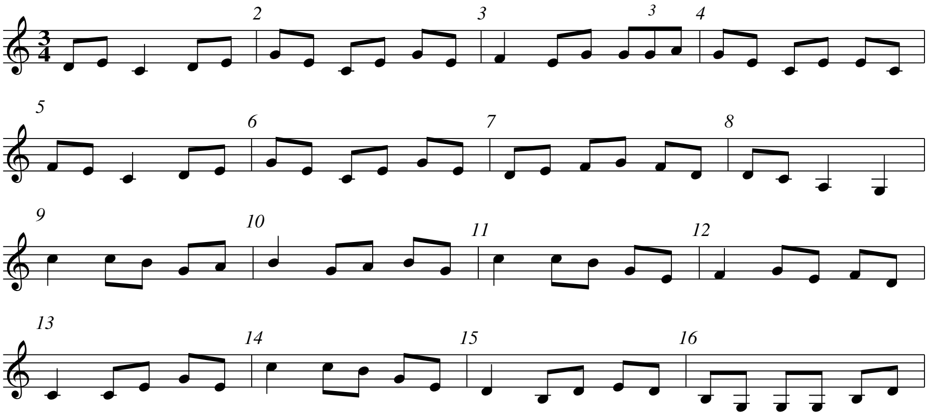

Fig. 7. Notation of transcription #3175, which can be heard here: goo.gl/7Fh5i4.

Fig. 8 shows transcription #7152, which can be heard here: `http://goo.gl/CaqRvr`. This tune has two eight-bar phrases in AABB form. In the A section, the variations of bb. 1–2 in bb. 3–4 and 5–6 create good coherence. The last two bars of A ascend higher than the rest, and present a plausible ending to that section. The B section also shows repetition and variation, but this pattern lasts only one bar instead of two, and in general the variations are only slight. Bar 9 is exactly repeated in b. 13, and is slightly varied in bb. 10, 11 and 14. Bar 12 is a refreshing break, but both ending bars sound static, unrelated, and do not give a convincing ending to the section. While both sections have an identifiable idea onto which a listener can latch, the link between them is weak.

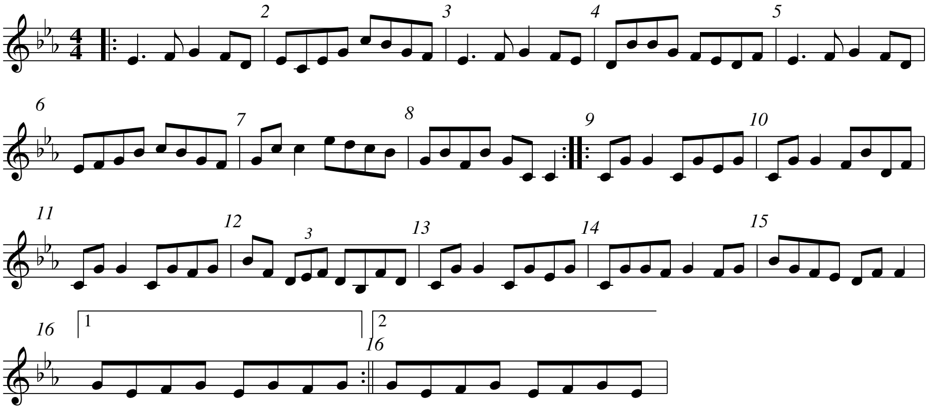

Fig. 8. Notation of transcription #7152, which can be heard here: goo.gl/CaqRvr.

Our analysis of these five randomly selected transcriptions generated by the *folk-rnn* model suggests that it has learned to some extent fundamental aspects of this kind of music: counting bars and accounting for pickup bars, stepwise motion in melody, and the pitches belonging to modes. The model also appears able to produce basic cadences, though these do not always work. We also see evidence of repetition and variation of musical ideas, which is an important aspect of such orally transmitted music, e.g., Boot, Volk, and de Haas (2016). The *folk-rnn* model also appears to have learned to some extent aspects of composing homophonic melodies, and assembling them into sections and larger forms with cadences occurring at appropriate times. Of these five transcriptions, four have an AABB form, with each section lasting eight bars, conforming to stylistic norms of the training material. The tunes are of varying quality, but overall we see plausible melodies that work.

At the same time, we find important aspects missing from these tunes. The harmonic implications of melodic patterns tend to be poor, which leads to weak or otherwise flawed cadences. Another missing element is the functional relationships between the different dimensions of the music. The *folk-rnn* model appears able to manipulate short melodic patterns and is also able to generate a conventional form, but is not able to relate these aspects as they are in the training data transcriptions. Similarly, the model appears able to sequence bars with correct note durations, but does not show an understanding of metre with its strong and weak beats in relation to melody and harmony. A melody will sound meandering when produced by manipulating patterns in a way that is disconnected from musical shape.

3.3 Nefarious Testing

Our results in the previous section suggest that the *folk-rnn* model has learned how to count bars, repeat and vary short patterns, and assemble them into larger-scale forms that resemble conventions of the training data. We now test the limits of the model in these directions by an approach we call "nefarious testing": we observe how the model behaves when we seed it with materials that are outside the conventions it has supposedly learned.

We first initialise the model with the following seed sequence: `<s> M:4/4 K:Cmaj [ G F ] 2 E G E /2 G > [ E F ] 4 |`, and have it generate a single transcription beginning with this bar (many outputs can be generated by changing the random seed, but we leave it to the default in every case). Our initial sequence is unusual with respect to the training transcriptions because it involves dissonant dyads, and an unconventional use of the `>` token.[8] We expect that if the *folk-rnn* model encodes musical knowledge about bar lengths, repetition and variation, and AABB form, then it will respond to our initialisation by generating a transcription having these basic conventions but using the initial material as the basis. The model outputs the transcription notated in Fig. 9. A synthesis can be heard here: `https://goo.gl/nP8cMr`.

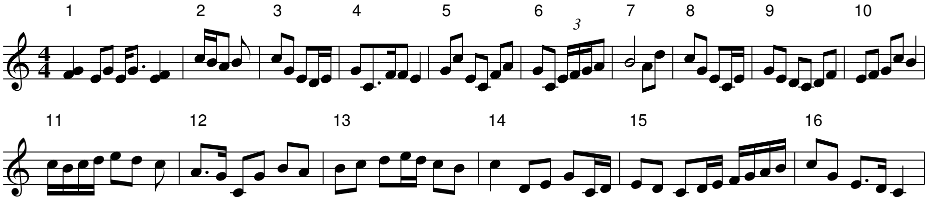

Fig. 9. Notation of the *folk-rnn* model output when initialised with the sequence `<s> M:4/4 K:Cmaj [ G F ] 2 E G E /2 G > [ E F ] 4 |`. A synthesis can be heard here: https://goo.gl/nP8cMr.

The *folk-rnn* model seems to ignore our initial idea since its output shows no repetition or discernible variation of it. We also see no bar has the correct duration with respect to our specified metre. After a bar of 3/8 it varies between 3/4 and 2/4, with one bar in 5/8. This shows that the model possesses a limited ability to count time correctly in a transcription. This is in contradiction to our observations in the previous section, as well as inspection of many other transcriptions generated by the model without initialisation. When it is not initialised, it correctly counts bars for the most part. One positive aspect of this generated sequence is that it ends with a cadence, though it is completely unexpected.

We now initialise the model with an equivalent bar, but expressed in a more conventional way: `<s> M:4/4 K:Cmaj [ G F ] 2 E G E < G [ E F ] 2 |`. The model outputs the transcription notated in Fig. 10. A synthesis can be heard here: `https://goo.gl/g3x5tn`.

[8] This ABC symbol is to denote a broken rhythm. Specifically, `>` means the note duration before is dotted and that after is halved.

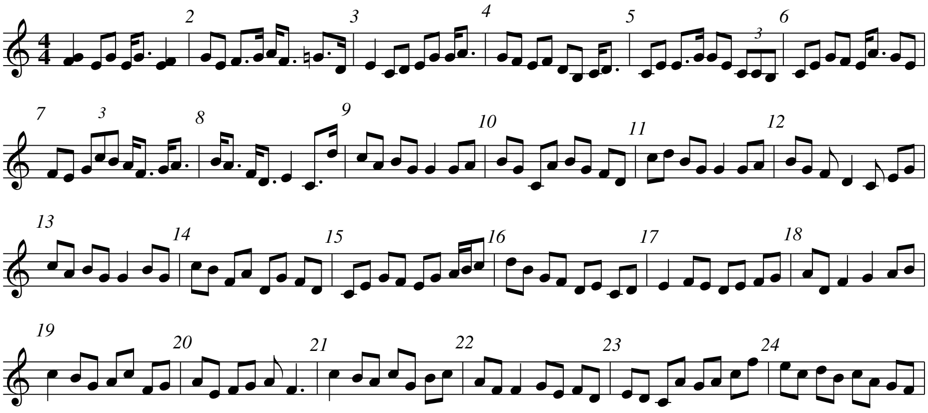

Fig. 10. Notation of the *folk-rnn* model output when initialised with the sequence `<s> M:4/4 K:Cmaj [ G F ] 2 E G E < G [ E F ] 2 |`. A synthesis can be heard here: https://goo.gl/g3x5tn.

We see that the model is still ignoring our initial idea, and produces an aimless transcription for many bars. A major difference with this output as compared to the first, however, is that the model now makes no mistakes in counting beats for the specified metre. It now seems able to count time. Another difference is that this output has a little more rhythmic consistency with respect to the initial bar, but this is promptly forgotten after b. 8. There appears to be some repetition and variation of ideas, e.g., bb. 9 and 13.

We now remove the dyads from this initialisation: `<s> M:4/4 K:Cmaj G 2 E G E < G F 2 |`. The *folk-rnn* model produces the transcription notated in Fig. 11. A synthesis can be heard here: `https://goo.gl/P1cKCc`.

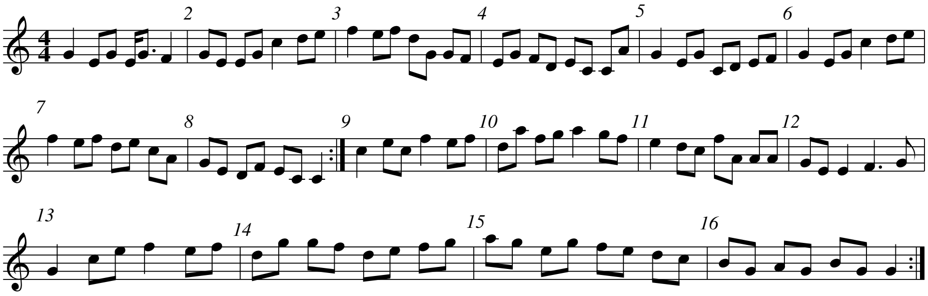

Fig. 11. Notation of the *folk-rnn* model output when initialised with the sequence `<s> M:4/4 K:Cmaj G 2 E G E < G F 2 |`. A synthesis can be heard here: https://goo.gl/P1cKCc.

The form of the output now resembles more closely that of the training material, with an AABB form, and cadences at appropriate points in A (bb. 4 and 8). We see more successful pattern repetition and variation, e.g., the GE idea in bb. 2, 5, 6, 8, etc., but the model still has not latched onto the rhythmic idea of the seed sequence.

Our nefarious testing of the *folk-rnn* model uncovers some intriguing limits to its musical knowledge that were not apparent with our other tests, or in the 30,000 transcriptions we had it generate. First, our assumption that the model can count bars appears true only in a very limited context. Second, our assumption that the model can repeat and vary material to create larger conventional forms also appears to be true only in a very limited context. As long as the initial sequence of tokens is similar enough to those in the training set – not only in terms of pitches but also rhythm – then the *folk-rnn* model can count bars correctly, repeat and vary material, and generate a conventional form with appropriately timed cadences. Breaking that similarity, however, shows that what the model has *actually* learned to do is not very general. This, of course, is rather different to what we reasonably expect of composition students, who are taught to generalise from specific cases to other domains, e.g., music styles. For example, learning how to apply music theory to understand baroque music also translates to applying it to understand popular music.

One might ask, what is a successful or acceptable continuation of our "nefarious" initialisation above? One possibility is (by Ben-Tal, notated for convenience) notated in Fig. 12. This can be heard here: `https://goo.gl/TVb6VE`. In this case, we recognise several characteristics of the initial bar and develop them in the continuation: the undulating third, the dyads, and the rhythm. The second bar reworks the initial idea, undulating on GE but in a different order, while the second half of the bar introduces a new rhythmic element, and a new pitch (A) while keeping the intervallic idea of the initial dyads. The third bar starts like the previous bar but here extends the undulation idea from two beats to three while circling on three pitches as in the initial bar. The three-dyad idea recurs (but back to crotchets) and leads into the next bar. Bars 4 and 5 return to the undulation on G and E, while adding E-flat as well and then the idea of the undulation and the dyad are mixed together in the crotchet triplet.

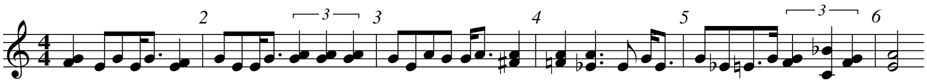

Fig. 12. Notation of one possible continuation of the sequence `<s> M:4/4 K:Cmaj G 2 E G E < G F 2 |`, composed by Ben-Tal. A synthesis can be heard here: https://goo.gl/TVb6VE.

Fig. 13 shows the notation of another possible continuation. This can be heard here: `https://goo.gl/ZYjpJE`. The two bars following the first have two beats that vary the rhythm of the undulation and two beats resembling the crotchets. The fourth bar returns to the minor third undulation but at a lower pitch level, and also extends this into a third beat. Bar 5 does the reverse: only the first beat relates to the undulation with the more static dyads taking three beats (and the final one arriving too soon, creating a surprise). Note also that the undulation and the dyads that were separate in the first bar become mixed, e.g. b. 3.

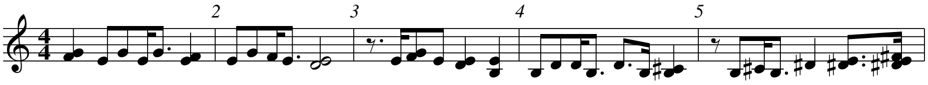

Fig. 13. Notation of another possible continuation of the sequence `<s> M:4/4 K:Cmaj G 2 E G E < G F 2 |`, composed by Ben-Tal. A synthesis can be heard here: https://goo.gl/ZYjpJE.

In both of these examples, Ben-Tal adds new pitches gradually (for the most part). In the first option, essentially one new note is introduced in each bar: A in b. 2; F-sharp in b. 3; E-flat in b. 4; b. 5 introduces two new notes: C and B-flat. In the second option, D is added in b. 2, and b is added in b. 3. These two notes then become the centre in b. 4, when C-sharp is added. This gradual drifting in pitch space, together with the transformation of the elements outlined above, produces a melody that is not just a variation on bar-level patterns, but that adds up to more then the sum of its parts. We must emphasise that the above discussion arises from *post-hoc* analysis, and do not accurately reflect the process of the composition. In other words, these were not conscious composition decisions, but only reflect aspects of the composer's approach to melodic writing and thinking about musical ideas and their development. This thus poses a problem if we want to fine-tune the *folk-rnn* model to become more responsive to a composer.

3.4 Assisted Music Composition in a Conventional Form

One practical aim of our work is to create models that facilitate music composition. In Sturm et al. (2016), we explored the interactive use of the *char-rnn* model for composing a piece of music that is outside the conventions of its training transcriptions. Now we explore using the *folk-rnn* model to interactively compose a conventional jig, which is a traditional form having an AABB structure in 6/8 metre. We evaluate the model *in situ* with the human composer. We are not interested in whether the model can compose music, but rather how it can contribute to the composition of music. We begin the process by initialising the model with a non-nefarious sequence, curate from its output, and seed anew with an expanded sequence.

Our seed sequence specifies the metre of a jig, the C-major mode, and a sequence of pitches known as the "Millennial Whoop":[9] `M:6/8 K:Cmaj G E G E 3`. We have the *folk-rnn* model generate three complete transcriptions. We reproduce below the command line and verbatim terminal output with line breaks added for readability. After each transcription we place a link to a synthesised realisation.[10]

```
python sample_rnn.py metadata/config5-wrepeats-20160112-222521.pkl \
--terminal --ntunes 3 --seed "M:6/8 K:Cmaj G E G E 3"
```

[9] See `https://goo.gl/TmFCQd`.

[10] To synthesise an ABC transcription, we convert it to MIDI, choose a subset of instruments (fiddle, accordion, guitar, harp, and whistle), and use *TiMidity* to create sound files. We also process the MIDI to create a percussion realisation to emulate a bodhran. We finally mix the sound files together.

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, D E C 2 A | G E D E C D | D A, G, A, 2 C :|
|: E G E E 2 E | C D E G 2 E | C E C A, 2 A, | B, D C A, B, G, |
E G E E 2 D | C E C A, 2 B, | C D E G E D |1 D 2 C C 2 D :|
|2 D 2 C C 2 E |
```

Realisation: https://goo.gl/NLGbo2

```
M:6/8
K:Cmaj
G E G E 3 | G E G c 3 | G E G c B A | G A G F E D |
G E G E 3 | G E G c 3 | d c A G A B | c B A G A B :|
c 3 d e d | c B A G A B | c B A B G A | A B c d e f |
c 2 d e d c | B A B c B c | A G F G A G | A B c d e d :|
```

Realisation: https://goo.gl/ew343V

```
M:6/8
K:Cmaj
G E G E 3 | G E G C 2 G | A G E G E C | D 3 E D C |
G E G E 3 | A G E G 3 | c B c d e f | e d c A 2 G :|
g e g g e g | a f a f e f | g e g g e g | a f d d 2 G |
g e g g e g | f a f a f e | g e c d 2 d | e c A G 2 G :|
```

Realisation: https://goo.gl/MaZNjH

All three transcriptions generated by the *folk-rnn* model have the form expected of a jig, and each repeats the Millennial Whoop in the fifth bar of each tune (A section).[11] The subject of the turn (B section) of the first and third transcriptions is a variation of the Millennial Whoop. The turn in the second output bears little relation to the tune. We find all three tunes, and two of the three turns, to be acceptable; but of them we prefer the tune of the first, and the turn of the last.

We take the tune of the first output as the initialisation for the model, and have it generate three alternative turns:

```
python sample_rnn.py metadata/config5-wrepeats-20160112-222521.pkl \
--terminal --ntunes 3 --seed "M:6/8 K:Cmaj G E G E 3 | A, C A, C 2 D | \
E G C E G c | B G A G E F | G E G E 3 | A, D E C 2 A | G E D E C D | \
D A, G, A, 2 C :|"
```

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, D E C 2 A | G E D E C D | D A, G, A, 2 C :|
|: E G c G 2 C | E G d e c G | E G c G 3 | G A c G E D |
E 2 c G 2 E | E G c A G E | D E D E D C | A, A, A, A, 3 :|
```

Realisation: https://goo.gl/tPWw9X

[11] The A section is called the “tune” and the B section the “turn” (Hillhouse, 2005).

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, D E C 2 A | G E D E C D | D A, G, A, 2 C :|
|: C D E G 2 E | C A, C G, 2 E | C D E G E G | A 2 A A G E |
C 2 E G 2 E | C A, G, G, E, G, | C D E G E C | D E D D 2 B, :|
```

Realisation: `https://goo.gl/wZScsk`

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, D E C 2 A | G E D E C D | D A, G, A, 2 C :|
G E G G E G | A G A c d e | d c A G A c | d c A G E G |
c d c d c A | G E C D E A | G E C D C A, | A, 3 G, 2 C :|
```

Realisation: `https://goo.gl/8pNbGq`

Of these, we prefer the first three bars of the turn of the first output, but with a modification such that its second and third bar are less static; i.e., we change `G | E G c G 3` to `A | E G c B 2 A`. This also inspires us to change the sixth bar of the tune from `A, D E C 2 A` to `A, C D C 2 A`. We use this as the new initialisation, and have the model finish the piece in three different ways:

```
python sample_rnn.py metadata/config5-wrepeats-20160112-222521.pkl \
--terminal --ntunes 3 --seed "M:6/8 K:Cmaj G E G E 3 | A, C A, C 2 D | \
E G C E G c | B G A G E F | G E G E 3 | A, C D C 2 A | G E D E C D | \
D A, G, A, 2 A :| |: E G c G 2 C | E G d e c A | E G c B 2 A |" \
--rng_seed 3213
```

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, C D C 2 A | G E D E C D | D A, G, A, 2 A :|
|: E G c G 2 C | E G d e c A | E G c B 2 A | G A A d c A |
G A E C D G | A G E D C A, | G, A, C D C A, | C D E C 3 :|
```

Realisation: `https://goo.gl/zknK8i`

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, C D C 2 A | G E D E C D | D A, G, A, 2 A :|
|: E G c G 2 C | E G d e c A | E G c B 2 A | G C C E C D |
E G c G C D | E G c A 2 B | c d c G E C | D E C A, 2 G, :|
```

Realisation: `https://goo.gl/RdsdjM`

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, C D C 2 A | G E D E C D | D A, G, A, 2 A :|
|: E G c G 2 C | E G d e c A | E G c B 2 A | E D D E G A |
E G c G 2 C | E G d c A G | E C A, G, A, C | D D D D 2 D :|
```

Realisation: `https://goo.gl/kFzAQp`

Notice that the command above involves reseeding the random number generator. We actually tried several random seeds until the model produced material that we found acceptable enough. Of these three, we prefer the turn of the last, save the ending. We thus take everything but the last bar of the turn of that last one, change the first two notes in the fourth bar of the turn – changing `E D D E G A` to `G E D E G A` – and have the model generate a turn with two endings, i.e., we add the token `|1` at the end of the initialisation:

```
python sample_rnn.py metadata/config5-wrepeats-20160112-222521.pkl \
--terminal --ntunes 3 --seed "M:6/8 K:Cmaj G E G E 3 | A, C A, C 2 D | \
E G C E G c | B G A G E F | G E G E 3 | A, C D C 2 A | G E D E C D | \
D A, G, A, 2 A :| |: E G c G 2 C | E G d e c A | E G c B 2 A | \
G E D E G A | E G c G 2 C | E G d c A G | E C A, G, A, C |1" \
--rng_seed 14
```

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, C D C 2 A | G E D E C D | D A, G, A, 2 A :|
|: E G c G 2 C | E G d e c A | E G c B 2 A | G E D E G A |
E G c G 2 C | E G d c A G | E C A, G, A, C |1 D E D D 2 E :|
|2 D E C D C D |
```

Realisation: `https://goo.gl/zEtYku`

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, C D C 2 A | G E D E C D | D A, G, A, 2 A :|
|: E G c G 2 C | E G d e c A | E G c B 2 A | G E D E G A |
E G c G 2 C | E G d c A G | E C A, G, A, C |1 E D C A, 3 :|
|2 E D C A, G, A, |
```

Realisation: `https://goo.gl/MDJWDk`

```
M:6/8
K:Cmaj
G E G E 3 | A, C A, C 2 D | E G C E G c | B G A G E F |
G E G E 3 | A, C D C 2 A | G E D E C D | D A, G, A, 2 A :|
|: E G c G 2 C | E G d e c A | E G c B 2 A | G E D E G A |
E G c G 2 C | E G d c A G | E C A, G, A, C |1 D E D C 2 D :|
|2 D E C C 2 B, |
```

Realisation: `https://goo.gl/iBpMjr`

As before, we produced the above from trying different random seeds. Of these three, we prefer the last one. We make a few final edits to the turn to finish "The Millennial Whoop Jig", notated in Fig. 14. We take a similar approach to compose "The Millennial Whoop Reel", which can be heard here: `https://goo.gl/9FWa1K`.[12]

[12] In this case, however, we had to add the Millennial Whoop manually, because the model could not be persuaded to repeat the motif. For further information, see `https://goo.gl/kLffNm`.

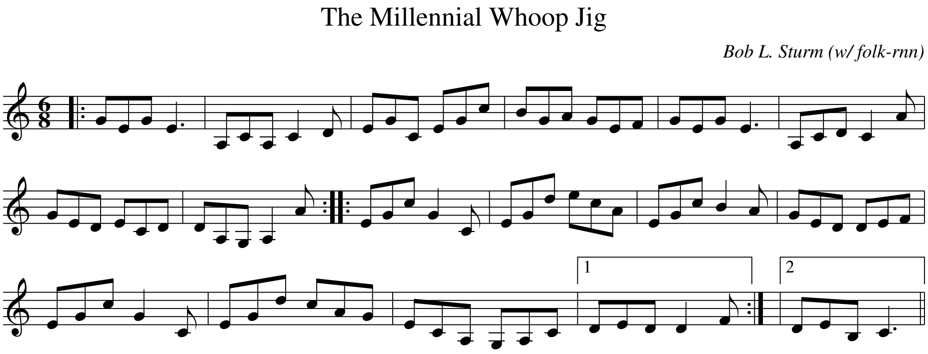

Fig. 14. The "Millennial Whoop Jig", a realisation of which can be heard here: https://goo.gl/nJeEsi.

3.5 Back to the Domain of Practice

To further gauge the plausibility of the transcriptions of *folk-rnn*, and to take the machine learning back into the practice domain from which its training data comes – the "machine learning that matters" of Wagstaff (2012) – we published online *The folk-rnn Session Book, Volume 1 of 10*, which contains 3,000 transcriptions with ABC notation, staff notation, and links to synthesised realisations.[13] We have invited comment on this collection from the session community via the website `http://thesession.org`. The introduction to the collection includes the following description:

> As the developers of this [music generation] system, we want to know what it has learned to do. What has it got right, and what has it got wrong? Since such information can greatly help us to improve the system, and adapt it for a variety of different uses, we are seeking input from people experienced in playing session music. Examples include:
>
> 1. How hard is it for you to find tunes in this volume that you think are close to the kind of music you encounter at a session? If you can, identify some tunes that fit this description, and explain why.
> 2. How hard is it for you to find tunes in this volume that you think are far from the kind of music you encounter at a session? If you can, identify some tunes that fit this description, and explain why.
> 3. Pick a tune in this volume that you think is close to the kind of music you encounter at a session and say how it could be improved, if at all.
> 4. Can you find a tune in this volume that is close to one that already exists?

The following selection from the comments thread[14] demonstrate some of the musical insights offered by engaging with such practitioners:

[13] This volume contains the first 3,000 transcription of the 30,000 we generate for the analyses in Sec. 3.1 and 3.2. For more information, see `https://goo.gl/qDESnL`.

[14] `https://thesession.org/discussions/39604`.

- *Kenny*: I had a look at the first three tunes, and listened to the third. The first two are garbage – there is none of the "question/answer" type of repetition within either tune which is the hallmark of traditional Irish dance music. The third one sounded a bit more convincing, but there are irregularities in a few of the bars.... I tried #6, but in D major, and not Dmix, and it almost works, although it opens very much like "The Floating Crowbar". Could work with a bit of "tweaking". Tune #4 – the jig – spends far more time up on "high c" than any other Irish tune I've ever come across.
- *Colman O'B*: I just had a first attempt at skimming through a bunch of the tunes and trying them on the whistle, and my overall feeling from the random tunes I picked is that if you go bar-by-bar there are lots of very "traditional sounding" phrases, but as Kenny says they tend not to actually go together to form coherent tunes as a whole. Tune 39 was actually sounding like quite a plausible traditional tune up until it ended on a low E when it clearly should have ended on a G instead.
- *Jim Dorans*: I just paged down to a random tune (actually #4) and I like it. If this is the result of an algorithm, it's doing pretty well.... I think it's good for its "austere" melody. I wouldn't bother transposing it – it works fine on fiddle. The high C is quite common in tunes – that said, the high C in bar #7 does sound a little bit odd – maybe drop that one down an octave? The other high Cs are fine, and they "go with the flow".
- *Alex Henry*: I found the work interesting, and I was surprised by how "Irish" the odd tune came out. I didn't even peruse what would be considered a valid sampling, but if I found one tune in the five I looked at to be good, it doesn't seem likely I found the one good tune in 3,000.
- *Conán McDonnell*: Some [tunes] were terrible – nice phrases now and again but without a proper resolution, and one or two were actually OK and sounded playable. Pot luck, really. I didn't find myself saying "these are great tunes" so much as "I could do something with that one", which I think is what really matters. The music comes from the player. A hundred people might play "Fishing For Eels", for example, but maybe only one or two play it in a way worth listening to. There are loads of crap tunes written by humans and there will continue to be (as long as there are people like me) in perpetuity. I'd rather play a great tune written by a computer than a crap one I wrote myself.

These comments confirm our own observation that the generated tunes mostly 'work' at the local level, but that their harmony and phrase structure often do not. Some comments also display a willingness to consider a generated transcription as a starting point to compose a tune. We see this with "The Mal's Copporim", a transcription generated by the *char-rnn* model (Sturm et al., 2016). This is in direct contrast to the user *Ergo*, who expounds a technophobic view: "Not only is there no point [to this research] but some 3,000 machine-generated tunes are now in the public domain, which troubles me.... My concern is that some people [may] consider one or more of these tunes [to] be actual traditional tunes.... I think it's reckless to send 3,000 machine-created fiddle tunes into the world". User *CreadurMawnOrganig* responds to *Ergo* with the observation that no matter what, humans will still be a part of the loop:

> Most tunes that we (traditional musicians, actual and aspiring) play, whether or not they have a known composer, have been shaped by the hands of many musicians – they have had the spiky bits rounded, the difficult bits made easier, the fussy bits made simpler, the boring bits made more interesting, the clunky

> bits made smoother and so on. If a robot-generated tune were successfully introduced into the session repertoire it would inevitably undergo the same process at the hands of us humans, ergo it ends up being (in part, at least) a human creation, anyway.

Building on this work, we invited professional musician and composer Torbjörn Hultmark[15] to select and perform some transcriptions from the volume of *folk-rnn* transcriptions as part of a concert of electronic and interactive music at the Centre for Digital Music at Queen Mary University of London (Nov. 2016). Hultmark is a classically trained trumpet player, but has performed for the past 20 years in various contexts, from jazz and free improvisation to electronic music. Hultmark essentially paged to near the middle of the *folk-rnn* volume, and looked closely at about ten transcriptions. He finally settled on performing three, with a choice of tempo and character that were intuitive and related to shaping the three into a coherent performance. He also used digital effects to alter his sound to add variation and interest to his performance. A video of his performance is available at `https://www.youtube.com/watch?v=4kLxvJ-rXDs`.

Hultmark had the following observations to offer after the exercise:

> Without exception the tunes are surprisingly catchy, easy and satisfying to play. 'Surprise' is probably an understatement here – I was bowled over by the playability of these tunes! Moreover, the more I played them the more I liked them. Each tune seemed very much to have its own character.

Of the ten transcriptions he looked at, he found the tunes convincing in general, though he pointed out that his familiarity with the style is limited. He also identified some problems connecting the sections – a lack of "coming to rest" at the end of phrases and awkward moments connecting the end of the phrase either back to the first bar or to the next phrase.

> I needed to make some (improvised) changes to the tunes in order to make it sound more convincing on my solo wind instrument, i.e., slow down at phrase-endings, omit notes, take more time at wide leaps, etc.

His final comment is encouraging as we take this project forward:

> The electronically generated 'Irish' tunes were a real revelation for me, and the collection has very much made me think/question a lot about the creative process. The experience has certainly made me look at music, art and composition in a different light.

Finally, two other events we organised in 2017 showcased both the *char-rnn* and *folk-rnn* models within the context of music performance, and demonstrate how the *folk-rnn* system is producing real impact outside the realms of the university. First, as part of the Inside Out Festival in London, we organised the workshop, "Folk music composed by a computer!?"[16] We invited participants to learn and discuss our research, learn a tune generated by the *char-rnn* model, and listen to a professional group of session players perform three sets of tunes.[17] Each set featured one computer-generated

[15] `http://www.hultmark.me/`.

[16] `goo.gl/YFNwAa`.

[17] Recordings of these can be heard here: `https://www.youtube.com/playlist?list=PLdTpPwVfxuXrjh41UZqcHACKWnsLttmIf`.

tune, two of which come from *The folk-rnn Session Book, Volume 1 of 10.* The lead musician involved with this workshop estimated that about one in five transcriptions in this volume are surprisingly good, and said that he had little trouble finding interesting tunes in the collection to include in the performance. The second event we organised was a public concert in London featuring music generated by and composed with the *char-rnn* and *folk-rnn* models, as well as other artificial systems.[18]

The innovative nature of this project attracted some media attention:

- June 3 2017, "An A.I. in London is Writing Its Own Music and It Sounds Heavenly", `https://goo.gl/zXhw1F`.
- June 18 2017, "Real Musicians Evaluate Music Made by Artificial Intelligence", `https://goo.gl/T6uVPC`.

These articles include the perspective of some of the musicians who performed in our concert as well as the audience, which expand the discussion beyond the common human-*versus*-machine narrative. The concert highlighted the crucial role performers play often neglected in discussion of creative music systems. At the same time the controversial nature of the project is reflected in ambiguous responses from some of the musicians and audiences.

4 Discussion

All five of our evaluation approaches attempt to illuminate what the *folk-rnn* model has actually learned to do, and determine the generality of its knowledge within the context of the symbol-sequences it has been trained to reproduce. We must emphasise that the model is not composing music; it is recursively generating a symbolic sequence according to joint probability distributions it has estimated from abstract and reductive representations of music crowd-sourced from a variety of music practitioners.

This motivates a comparison of the descriptive statistics of transcriptions generated by the model and its training transcriptions. From this we can identify similarities and differences between the two populations. There are clear biases, but we see that the model still seems to have learned enough about the training data such that it can generate new transcriptions that are plausible with respect to some basic conventions of the music practice. This motivates us to analyse the output of the model as one would that of a composition student, allowing us to cautiously evaluate what "musical" knowledge it may have learned. We find that the model appears to have learned aspects of pattern repetition and variation, of constructing sensible forms with some amount of consistency, and even of using functional elements such as cadences. We can only gain so much insight into the generality of the model's knowledge by inspecting a collection of transcriptions it generates without directives. We thus interrogate it using seed sequences that diverge from the training material, and thereby find several limitations to what the model appears to be doing. This "nefarious testing" shows that the model can count and repeat and vary patterns, but only within a very restricted context. It is interesting that an uncommon use of the token `>` dramatically confuses the model. On the one hand, this reveals that it is responding to context, and not just to individual items in a sequence, which is necessary when trying to model music. On the other hand, it becomes clear that the model has not learned the function of the token `>`, which calls into question its "understanding" of other ABC tokens.

[18] Recordings of pieces of the program can be heard here: `https://goo.gl/ZdGtwM`.

One may remark of our evaluation that we have omitted testing whether people can detect the difference between a transcription composed by a human and one generated by the *folk-rnn* model, often erroneously called a “music Turing test” (Ariza, 2009). We are not interested in creating a music composition machine that can fool people into thinking its output must have been created by a human. That narrative – human *versus* machine – might be accessible and entertaining, but it is accompanied by a danger of trivialising the endeavour, and misunderstanding it in a threatening way.[19] Furthermore, any significance of having accomplished such “fooling” is ultimately weak and uninformative: Who was fooled, and why? The “lay listener” or an “expert”? Was an obscure “real” transcription picked? Was a particular machine-generated output cherry-picked? Why should one not cherry-pick from the output of a machine when people cherry-pick from human composers all the time? In the end, we do not find this approach to be useful to improving the system, or measuring its usefulness in music practice.

One may also remark that the approaches to evaluation we use are by and large unsystematic. Comparing the statistics of transcription populations may be systematic, but it is far removed from the use of a model of music co-creation, and the usefulness of the output. Generating thousands of transcriptions, and randomly selecting from these a subset for closer inspection may be systematic, but the subsequent musical analysis of each transcription is not. Since the aim of our research is not strictly style emulation, but in the use of a model for music co-creation, we have instead looked at measuring the potentials and limitations of the system to be of real use to practitioners. Adding to the complexity of evaluation is the fact that even if the model fails in its output with regards to a specific domain of music practice, this can bring about new creative inspiration.[20] We see a multifaceted approach to evaluation as the best option for such a use case.

In a rough way, the *folk-rnn* model embodies a “folk tradition” that one can use for inspiration and musical ideas, much like classical composers have done with real folk traditions, e.g., Haydn, Bartók, Sibelius, Liszt, Grainger, Robert Nathaniel Dett, George Crumb and Beth Anderson. Other creative possibilities come from sampling from the output layer of the neural network in less conservative ways. This results in transcriptions that are far from those of session music, with some examples looking like parodies of “new music”. For example, sampling from the posterior distribution at the output of the model using a higher “temperature” – thereby flattening the posterior distribution – generated the following sequence:

```
<s> M:4/4 K:Cmin c d e 2 B, =B, F, ] =A _B, =c ^C ] _b ^c’ ^d A 8 =B =C \
|: b E, (2 M:12/8 2 > (3 B, ^F, A, _g [ <s> =f =G =A =c =B, ^c M:3/4 \
(3 _D G, </s>
```

This is notated in Fig. 15. Such results are often incorrect ABC, e.g., in the example above there are unmatched chord tokens `]`, spurious transcription tokens `<s>`, an incorrect grouping token `(2`, and incorrectly counted bars. Nevertheless, such approaches to generation can provide material from which to draw new ideas and inspiration.

Regardless of what the *folk-rnn* model has *actually* learned to do, we find its use for “augmenting” music practice very promising. Several practitioners have admitted

[19] For instance, see the comments by *Ergo* in Sec. 3.5.

[20] For a demonstration, see Sturm’s composition “Eight Short Outputs . . . ”, `https://youtu.be/RaO4HpM07hE`.

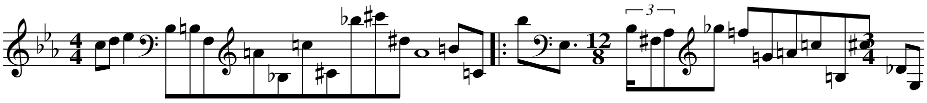

Fig. 15. Notation of *folk-rnn* output when sampling at a high temperature.

surprise by the plausibility of transcriptions generated by the model, and are now participating in evaluating the generated tunes within a wide variety of musical practices. For our own compositional work, both *char-rnn* and *folk-rnn* continue to provide effective means for generating or suggesting material for music composition. Our co-creation with the model has led to the composition of a growing number of new works:

1. "Bastard Tunes" by Oded Ben-Tal + folk-rnn (2017) `https://goo.gl/qqXXYj`
2. "Chicken Bits and Bits and Bobs" by Bob L. Sturm + folk-rnn (2017)
   `https://youtu.be/n-avS-ozrqU`; score: `https://goo.gl/LHxtfh`
3. "March to the Mainframe" by Bob L. Sturm + folk-rnn (2017)
   `https://youtu.be/TLzBcMvl15M`; score: `https://goo.gl/37B2sG`
4. "Interlude" by Bob L. Sturm + folk-rnn (2017)
   `https://youtu.be/NZO8dDdYh3U`; synthesized version: `https://goo.gl/HFmA32`; score: `https://goo.gl/z5hVsp`
5. "The Humours of Time Pigeon" by Bob L. Sturm + folk-rnn (2017)
   `https://youtu.be/1xBisQK8-3E`; synthesized version: `https://goo.gl/n9kDxk`; score: `https://goo.gl/EGtWEL`
6. "The Glas Herry Comment" by folk-rnn + DeepBach (2017)
   `https://youtu.be/y9xJl-ljOuA`
7. X:7153 by folk-rnn + DeepBach (2017) `https://youtu.be/tdKCzAyynu4`
8. X:633 by folk-rnn + DeepBach (2017)
   `https://youtu.be/BUIrbZS5eXc`
9. "Optoly Louden" by folk-rnn + Bob L. Sturm (2017)
   `https://youtu.be/BaRw01c76PA`
10. X:488 by folk-rnn (2017)
    `https://youtu.be/QWvlnOqlSes` and `https://youtu.be/QWvlnOqlSes`
11. "The Fortootuise Pollo" by Bob L. Sturm + folk-rnn (2017)
    `https://goo.gl/92WD6L`
12. "It came out from a pretrained net" by Bob L. Sturm + folk-rnn (2016)
    `https://goo.gl/EBEvbn`
13. "The Ranston Cassock" by Bob L. Sturm + folk-rnn (2016)
    `https://youtu.be/JZ-47IavYAU`,
    version for viola and tape: `https://goo.gl/9CrKUg`
14. "The Millennial Whoop Reel" by Bob L. Sturm + folk-rnn (2016)
    `https://goo.gl/1jRhvW`
15. "The Millennial Whoop Jig" by Bob L. Sturm + folk-rnn (2016)
    `https://goo.gl/vfvBqE`
16. "Eight Short Outputs . . . " by folk-rnn + Bob L. Sturm (2015)
    `https://youtu.be/RaO4HpMO7hE`
17. "The Drunken Pint" by folk-rnn (2015)
    `https://youtu.be/omHhyVD3PD8`, `https://youtu.be/0gosLln8Org`

18. “The Glas Herry Comment” by folk-rnn (2015) https://youtu.be/QZh0WSjFFDs, https://youtu.be/NiUAZBLh2t0
19. “We three layers o’ hidd’n units are” by Bob L. Sturm + folk-rnn (2015) https://goo.gl/jF6Dvc
20. “The March of Deep Learning” by Bob L. Sturm + folk-rnn (2015) https://goo.gl/SG3tMe
21. “The Mal’s Copporim” by folk-rnn (2015) https://youtu.be/YMbWwU2JdLg, https://youtu.be/HOPz71Bx714
22. “The Castle Star” by folk-rnn + Bob L. Sturm (2015) https://goo.gl/CSm1fX
23. “The Doutlace” by folk-rnn + Bob L. Sturm (2015) https://goo.gl/jsLmKj

5 Conclusion

The immediate practical aim of our work is to create models of music transcriptions that facilitate music composition and practice, both in and outside particular conventions (Sturm et al., 2016). This article concludes our exploration of two approaches to modelling session music transcriptions, and demonstrates several approaches to evaluating the models, including taking them back into the domain of practice (Wagstaff, 2012). We find that both approaches are able to capture stylistic conventions exhibited by the training data, and that both models can be useful for creating new music, in and out of those conventions – but with plenty of caveats. We aim to improve these models to make them more responsive to the input of a composer; to make their knowledge more adaptable to the style specific to a composer or other practice; and to facilitate more approaches to composition than just generating material in a left-to-right fashion. For example, the current implementation of *folk-rnn* does not produce models that can generate material occurring between two given sequences. Making *folk-rnn* models more flexible tools for composition will require adding meaningful controls with which a composer can dynamically shape the generation process. Another avenue that we are exploring is the integration of a “critic” in the generation loop, and the application of reinforcement learning to tune the model (Jaques et al., 2017). Such approaches can serve either to smooth out some of the problems with generated material, or could serve as a basis for a composer to define an individual aesthetics that will steer the composition process. Another approach would be to include some additional controls at the generation process. For this approach to be most effective we would need to understand better how the model learned what it did, how this knowledge is encoded, and how that knowledge can be changed in predictable ways.

References

Ariza, C. (2009). The interrogator as critic: The Turing test and the evaluation of generative music systems. *Computer Music Journal*, *33*(2), 48–70.

Boot, P., Volk, A., & de Haas, W. B. (2016). Evaluating the role of repeated patterns in folk song classification and compression. *Journal of New Music Research*, *45*(3), 223–238. doi: 10.1080/09298215.2016.1208666

Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In A. McCallum, K. Weinberger, &

A. Globerson (Eds.), *Proc. int. conf. machine learning* (pp. 1159–1166). Edinburgh, Scotland: International Machine Learning Society.

Chen, C. J., & Miikkulainen, R. (2001). Creating melodies with evolving recurrent neural networks. In K. Marko & P. Werbos (Eds.), *Proc. int. joint conf. neural networks* (pp. 2241–2246). Honolulu, Hawaii: IEEE.

Choi, K., Fazekas, G., & Sandler, M. (2016). Text-based lstm networks for automatic music composition. In *Proc. 1st conference on computer simulation of musical creativity.* Huddersfield, UK. Retrieved from `https://drive.google.com/file/d/0B1OooSxEtl0FcG9MYnY2Ylh5c0U/view`

Colombo, F., Muscinelli, S. P., Seeholzer, A., Brea, J., & Gerstner, W. (2016). Algorithmic composition of melodies with deep recurrent neural networks. In *Proc. 1st conference on computer simulation of musical creativity.* Huddersfield, UK. Retrieved from `https://drive.google.com/file/d/0B1OooSxEtl0FYWxidUNFS3V2TVk/view`

Deng, L., & Yu, D. (2014). *Deep learning: Methods and applications.* Redmond, WA: Now Publishers.

Eck, D., & Lapamle, J. (2008). *Learning musical structure directly from sequences of music* (Tech. Rep.). Montreal, Canada: University of Montreal.

Eck, D., & Schmidhuber, J. (2002). Learning the long-term structure of the blues. In *Proc. int. conf. artificial neural networks* (pp. 284–289). Madrid, Spain: LNCS.

Franklin, J. A. (2006). Recurrent neural networks for music computation. *Journal on Computing*, *18*(3), 321–338.

Hillhouse, A. N. (2005). *Tradition and innovation in Irish instrumental folk music* (Unpublished master's thesis). The University of British Columbia.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., ... Sainath, T. N. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine*, *29*(6), 82–97.

Jaques, N., Gu, S., Turner, R. E., & Eck, D. (2017). Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In T. Jebara, D. Roy, & I. Murray (Eds.), *Proc. int. conf. machine learning.* Sydney, Australia: Journal of Machine Learning Research.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Proc. neural info. process. syst.* (pp. 1097–1105). Lake Tahoe, USA: Neural Information Processing Systems Foundation, Inc.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Loughran, R., & O'Neill, M. (2016). Generative music evaluation: Why do we limit to 'human'? In *Proc. 1st conference on computer simulation of musical creativity.* Huddersfield, UK. Retrieved from `https://drive.google.com/file/d/0B1OooSxEtl0FMk0tbm9PWjl4NEk/view`

Mozer, M. C. (1994). Neural network composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing. *Cognitive Science*, *6*(2&3), 247–280.

Pearce, M., Meredith, D., & Wiggins, G. (2002). Motivations and methodologies for automation of the compositional process. *Musicae Scientiae*, *6*(2), 119–147.

Sturm, B. L., Santos, J. F., Ben-Tal, O., & Korshunova, I. (2016). Music transcription modelling and composition using deep learning. In *Proc. 1st conference on computer simulation of musical creativity.* Huddersfield, UK. Retrieved from `https://drive.google.com/file/d/0B1OooSxEtl0FcTBiOGdvSTBmWnc/view`

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, & K. Weinberger (Eds.), *Proc. neural information process. systems* (pp. 3104–3112). Montréal, Canada: Neural Information Processing Systems Foundation, Inc.

Todd, P. M. (1989). A connectionist approach to algorithmic composition. *Computer Music Journal*, *13*(4), 27–43.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., … Kavukcuoglu, K. (2016, September). WaveNet: A Generative Model for Raw Audio. *ArXiv e-prints (1609.03499).*

Wagstaff, K. L. (2012). Machine learning that matters. In A. McCallum, K. Weinberger, & A. Globerson (Eds.), *Proc. int. conf. machine learning* (pp. 529–536). Edinburgh, Scotland.