StructureNet: INDUCING STRUCTURE IN GENERATED MELODIES

Gabriele Medeot[1] **Srikanth Cherla**[1] **Katerina Kosta**[1] **Matt McVicar**[1]
Samer Abdallah[1] **Marco Selvi**[1] **Ed Newton-Rex**[1] **Kevin Webster**[2]
[1]Jukedeck Ltd., London, United Kingdom
[2]Imperial College London, London, United Kingdom
{gabriele, srikanth, katerina, matt, samer, marco, ed}@jukedeck.com
kevin.webster@imperial.ac.uk

ABSTRACT

We present the StructureNet - a recurrent neural network for inducing structure in machine-generated compositions. This model resides in a musical structure space and works in tandem with a probabilistic music generation model as a modifying agent. It favourably biases the probabilities of those notes that result in the occurrence of structural elements it has learnt from a dataset. It is extremely flexible in that it is able to work with any such probabilistic model, it works well when training data is limited, and the types of structure it can be made to induce are highly customisable. We demonstrate through our experiments on a subset of the Nottingham dataset that melodies generated by a recurrent neural network based melody model are indeed more structured in the presence of the StructureNet.

1. INTRODUCTION

Automated generation of symbolic music using computers involves the application of computer algorithms to the creation of novel musical scores. The natural predisposition of computers to quickly enumerate and choose from a large set of compositional alternatives makes them suitable candidates for discovering novelty in the vast space of musical possibilities that could be daunting to a human composer. Leveraging computing power for this purpose has the potential to aid and accelerate the creative process, thus lowering the bar for composition and democratising it. So-called *machine-generated music* has been a subject of steady interest since the pioneering work of a few musically inclined information theorists [5, 8]. This interest has surged during the past decade or so within academia and especially outside it with the rise of certain industry players (such as Jukedeck [1] and the Magenta project [2]).

The roughly seven decade-long history of machine-generated symbolic music has seen the application of a plethora of algorithms to varying degrees of success [8]. With the increasing digitisation of musical scores, those relying on machine learning have gained importance in recent times. The relatively successful approaches among these have been Probabilistic Grammars [9], (Hidden) Markov models [19, 21], and Connectionist architectures [2, 18]. The latter in particular have proven to be highly effective at representing musical information and modelling long-term dependencies which are crucial to generating good-quality music [3].

This paper addresses the issue of long-term structure in machine-generated symbolic monophonic music. Structure is a key aspect of music composed by humans that plays a crucial role in giving a piece of music a sense of overall coherence and intentionality. It appears in a piece as a collection of musical patterns, variations of these patterns, literal or motivic repeats and transformations of sections of music that have occurred earlier in the same piece. Hampshire underlines that a piece can be conceived as a work of art if and only if the listener's mind is actively tracing the structure of the work using her own natural imagery and musical memory [7, p. 16].

Here we introduce StructureNet - a recurrent neural network that induces structure in machine-generated melodies. It learns about structure from a dataset consisting of structural elements and their occurrence statistics, which is created using a structure-tagging algorithm from an existing dataset of melodies. Once trained, StructureNet works in tandem with a melody model which generates a probability distribution over a set of musical notes. Given the melody model's prediction at any given time during generation, StructureNet uses the structural elements implied by the melody so far to alter the prediction, leading to a more structured melody in the future. Our experiments reveal that music generated with StructureNet contains significantly better structure, even when it is trained on a relatively small dataset. We provide musical examples that highlight this fact.

The next section introduces relevant state-of-the-art. Some preliminaries and a description of StructureNet follow in Sections 3 and 4 respectively. Based on the results presented in Section 5, we summarise our findings and suggest potential future work in Section 6.

[1] https://www.jukedeck.com/

[2] https://magenta.tensorflow.org/

2. RELATED WORK

In order to repeat verbatim or with variations sections that have occurred previously in a piece of machine-generated music, i.e. to induce structure in it, the model must be able to encode and recall in some way what has happened in the past. This can be achieved in a variety of ways. In a first instance, improving structure simply involves making the generation model more powerful. An example of this is the RNN-RBM [2] that was enhanced purely by replacing its components - the Recurrent Neural Network (RNN) by a Long-Short Term Memory (LSTM) Network to improve its temporal memory, making it the LSTM-RTRBM [16], and the Restricted Boltzmann Machine (RBM) by a Deep Belief Network (DBN) to improve its output layer, making it the RNN-DBN [10]. Similarly, it was demonstrated in [4] that connectionist models outperform Markov models in modelling melodic sequences. Closely related to these is a musically informed improvement that enriches the feature encoding to include those features that have the potential to add more information about structure [6, 19]. Along similar lines, the Magenta Project proposed two neural network architectures to model higher-level structure in music - the Lookback RNN and Attention RNN [25]. While the former augments the model's feature vector with information about notes from previous measures, repeat information and metrical location, the latter adopts an attention-based mechanism [1] wherein a weighted sum of the model's outputs in the previous n locations is used in addition to its current state to make better predictions. Such approaches address the overall quality of music, of which high-level structure is just one aspect. Moreover, the improvements afforded by the former kind are highly dependent on the training loss, which does not explicitly take into account structure of the kind observed in music. So while an improvement in the model or feature representation does tend to improve the overall quality of music in a piece, improvement is often observed over short time-spans and not necessarily in the higher-level structure.

Alternatively, one can explicitly addresses the issue of high-level structure in machine-generated compositions. One simple solution involves dividing the generation task between multiple models. The MELONET system [13], whose goal is to produce variations of a given melodic theme, achieves structural coherence by dividing the effort between two mutually interacting neural networks operating at different time-scales. The first network learns to recognise musical structure while the second network predicts the musical notes. Similarly, Todd [24] proposed two cascaded networks that allow the explicit representation of structure in a hierarchy. The first network generates a sequence of plans which correspond to descriptions of melodic chunks, and the second a sequence of notes given a plan. More recently, Roig et al. [22] devised a system in which melodic and rhythmic patterns existing in the dataset are concatenated according to statistically governed rules to form new patterns that are not too distant from those occurring in the dataset. In the system known as MorpheuS [11] music generation is formulated as a combinatorial optimisation problem in which a template of musical structure acts as a hard-constraint, and solved using a meta-heuristic search algorithm known as Variable Neighbourhood Search. Patterns contained in the dataset of pieces are discovered using an existing pattern-detection algorithm [17]. In a similar vein, [20] control the generation of chord sequences and melodies using steerable constraints Markov chains. Lattner et al. [14] adopt a similar approach where a Convolutional Restricted Boltzmann Machine is combined with a constraint optimisation technique to constrain the music sampled from the C-RBM according to the musical structure of a given template.

3. BACKGROUND

StructureNet is a Recurrent Neural Network (RNN) that operates in the space of musical structure and learns sequences of features that denote the presence or absence of repeats at a point in time and their type, if present. Here we give an overview of the Long Short-Term Memory (LSTM) RNN that underlies StructureNet and the definition of structural repeats that we rely on.

3.1 Long Short-Term Memory

The RNN is a type of neural network for modelling sequences and its basic architecture consists of an input layer, a hidden layer and an output layer. The *state* of its hidden layer acts as a memory of the past information it encounters while traversing a sequence. At each location in the sequence, the RNN makes use of both the input and the state of its hidden layer from the previous location to predict an output. Here we use a special case of the RNN known as the Long-Short Term Memory (LSTM) network [12] that, owing to the presence of purpose-built *memory cells* to augment its hidden layer, boasts a greater temporal memory than the standard RNN. Given an input vector x_t at sequence location t, the output of the LSTM h_{t-1} and its memory cell c_{t-1} (collectively, its state) from the previous location, the output of the LSTM layer h_t is computed and further propagated into another layer of a larger model.

3.2 Modelling Melodies and Structure Elements

The output layer of the note-based (as opposed to frame-based) melody model in the present work contains two groups of softmax units. Each group of softmax units models a single probability distribution over a set of mutually exclusive possibilities. The first of these denotes the musical pitch of the note, and the second its duration. Given the output of the LSTM layer h_t at any given location t in the sequence, this is transformed into two independent probability distributions $\mathfrak{p}_t$ and $\mathfrak{d}_t$ that together make up the output layer of the network. From these two distributions, the probability of a certain note (with pitch and duration) can be obtained simply by multiplying the probabilities of its corresponding pitch and duration respectively. Note that the output layer of StructureNet contains three groups of softmax units. Although these represent different quantities that define aspects of structure (explained in detail in

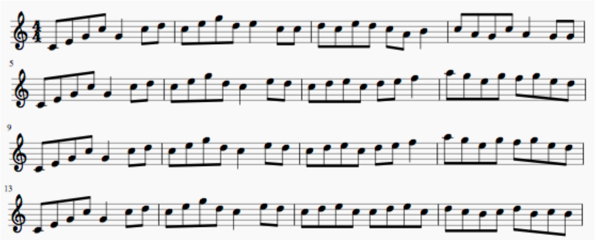

Figure 1. A 16-measure melody generated by our LSTM melody model together with the StructureNet. A selection of repeats in this melody are as follows: measures 9-12 are a duration-interval repeat of measures 5-8, as are measures 13-14 of measures 9-10; and measures 15 and 16 are both duration repeats of measure 12.

Sections 4.1 and 4.2), the manner in which these are combined to generate the probabilities of structural elements is identical to the melody model. Also note that the choice of the LSTM as the melody model is arbitrary and it can be replaced by any other probabilistic prediction model.

3.3 A Definition of Structure

There are various types of structure present in music. Composers use techniques such as instrumental variation, changes and repeats in timbre, and dynamics to induce a feeling of familiarity in the listener. In the present work, however, we focus on the score-level repeat information. In a score, perhaps the two most obvious types of repeat are of (1) duration (rhythmic), and (2) pitches (melodic).

A duration repeat is a section of the melody, the durations of whose notes are the same as those of a previous section. Examples of duration repeats can be found in the melody of Figure 1. These are determined purely by the sequences of crotchets and quavers contained in these measures. When it comes to pitch, it is helpful to think of these repeats in terms of *intervals rather than absolute pitch*. The interval between two notes can be defined in a number of ways, but in this work we use the scale degree distance between notes. For instance, in the key of C major, the scale degree between a C note and subsequent E note would be the same as the scale degree between a D note and subsequent F note. Given this definition of an interval, a duration-interval repeat is a section of the melody that holds the same relationship to a previous section as a duration repeat, and additionally the intervals between whose consecutive notes are the same as those between the consecutive notes of that previous section. Figure 1 also illustrates duration-interval repeats. In the present work, we consider duration repeats as well as repeats of both durations and intervals. Purely interval repeats were found to be very few in our chosen dataset and were thus ignored.

4. StructureNet

StructureNet is only able to produce structural repeat information that biases the predictions of an accompanying music (in the present case melody) model. In Section 4.2 we will outline a methodology whereby it modifies the probability of notes that the melody model produces, thus encouraging structure but not enforcing it. Crucially, this means that the structure network is able to *suggest* repeats of certain types, but if the melody network assigns very low probability to notes that would form these repeats, it is free to "override" the structure network's suggestions in a probabilistic and flexible manner. The specifics of how StructureNet achieves these goals is outlined in the remainder of this section, beginning with the type of structure we capture and how we identify it.

4.1 A Dataset of Structure

StructureNet operates in a space of musical structure. In order to train the model, we first create this structure dataset by processing a dataset of melodies with a musical repeat-detection algorithm. The algorithm encodes each melody into a sequence of binary feature vectors in the semi-quaver temporal resolution (although this resolution is not a strict requirement: if the dataset contains no notes shorter than a quaver, one may use a quaver as the minimal resolution). The feature vector itself is a concatenation of three one-hot sub-vectors. The first is given by

$$[f, d, di_{tr}, di_{nt}]$$

wherein each bit of the first sub-vector indicates which of four categories a given frame of music belongs to. These are (1) f - free music, (2) d - duration repeat, (3) di_{tr} - duration-interval repeat with transposition and (4) di_{nt} - duration-interval repeat without transposition. The only distinction between the two types of duration-interval repeats is that in the case of di_{tr} the section to which the frame belongs is a transposed version of the original section whereas in the case of di_{nt} the section to which the frame belongs is at the same musical pitch as the original section. The free music bit f indicates that the frame is a part of a section that is not a repeat of any previous section of the melody. The second one-hot sub-vector is given by

$$[f, l_{0.5}, l_{0.75}, l_{1.0}, l_{1.5}, l_{2.0}, l_{3.0}, l_{4.0}, l_{8.0}, l_{16.0}]$$

and contains bits that indicate the *lookback*, i.e. the distance (in crotchets) between the original section and the section containing the current frame, if the section containing the current frame is a repeat of the original section. If it is not a repeat, the free music bit f is on. Note that the value of the free music bits in both sub-vectors is identical and hence uses the same notation. Also note that the choice of the set of lookbacks is completely open to change, and highlights another flexible aspect of the model; it may even be possible to learn the optimal set of lookbacks for a given dataset. Finally, the third 8-dimensional one-hot vector ϕ encodes the location of a frame via its *beat strength* β and its *measure strength* ρ. The beat strength [15] encodes the strength of each metrical location in a measure. In a measure divided into 16 semi-quaver beats (as in the present work), its values are $\beta = [0, 4, 3, 4, 2, 4, 3, 4, 1, 4, 3, 4, 2, 4, 3, 4]$. The measure strength extends the notion of beat strength to a sequence of measures. The strengths associated with beats in a measure are associated with measures in a piece, beginning with the first measure. In the present work, we choose a cycle of 8 measures that correspond to the following sequence of measure strengths $\rho = [0, 3, 2, 3, 1, 3, 2, 3]$. In both cases, a lower value indicates a higher strength. Our encoding is defined as

$$\phi_t = \begin{cases} \rho(\mathrm{mod}(t, 8)) & \text{if } \mathrm{mod}(t, 16) = 0 \\ \max(\rho) + \beta(\mathrm{mod}(t, 16)) & otherwise \end{cases} \tag{1}$$

Note that just as the beat strength, the measure cycle duration for the measure length can also be varied as desired.

StructureNet models the vector that is a concatenation of these three sub-vectors as three groups of softmax units in its output layer. As noted earlier in Section 3.2, the manner in which one combines the probability distributions represented by these two groups of softmax units (for instance, a duration-interval repeat of lookback 8.0, or an interval repeat of lookback 1.5) is by multiplying the corresponding probabilities one from each group.

The repeat-detection algorithm works by first converting a sequence of notes into two strings - one corresponding to durations and the other to intervals. In each of these strings, it then uses a string matching algorithm to find substrings that repeat. Single-note repeats are trivial and thus discarded, and only those repeats that correspond to the above listed lookbacks are retained. Any note that is longer than 2 measures is split into multiple notes of the same pitch to limit the number of characters required to represent the piece as a string. Then the list of duration repeats are filtered such that only the longest repeats remain and all overlapping and shorter repeats are discarded. At this stage, the duration-interval repeats are nothing but duration repeats with coinciding interval repeats. So from the list of interval repeats only those are retained that coincide exactly with the current list of duration repeats with the same lookbacks. These are tagged as duration-interval repeats, replacing the corresponding duration repeats to give the final list of duration repeats and duration-interval repeats. While it is indeed possible to look for other types of repeats, we limit ourselves in this paper to the above as it is sufficient to demonstrate the efficacy of StructureNet. This also highlights the flexibility of the model wherein one may change the type of repeats detected and also customise the number of lookbacks as needed.

4.2 Influencing Event Probabilities

Once trained on the above described structure dataset, StructureNet is then put to use with the probabilistic melody prediction model $\mathcal{M}_m$. At time t (that is, given the history of notes generated up to time t), the model $\mathcal{M}_m$ predicts a probability distribution P_t over a set of notes N. At the same time, given the history of repeats generated so far, the structure model $\mathcal{M}_s$ predicts a probability distribution Q_t over a set of possible repeats Π, which includes an element π_f, representing ‘free music’. Each note $\nu \in N$ can be consistent with a subset Π_t^ν of these repeats, which will always include π_f, meaning that every note is consistent with ‘free music’.

StructureNet influences the prediction P_t by modifying the probability of each note according to the probabilities of the repeats with which it is consistent. Let $\phi_t : N \times \Pi \rightarrow \{0, 1\}$ be a function such that $\phi_t(\nu, \pi) = 1$ when note ν is consistent with repeat π at time t and 0 otherwise. In terms of this we can express Π_t^ν as $\{\pi \in \Pi | \phi_t(\nu, \pi) = 1\}$, and further define $N_t^\pi = \{\nu \in N | \phi_t(\nu, \pi) = 1\}$, which is the set of notes consistent with π. Each note ν is then assigned a weight

$$W_t(\nu) = P_t(\nu) \sum_{\pi \in \Pi_t^\nu} \frac{Q_t(\pi)}{\mu_t^\pi}, \tag{2}$$

where $\mu_t^\pi = \sum_{\nu \in N_t^\pi} P_t(\nu)$. In this way, the relative probability of a note ν is increased when it is consistent with repeat(s) to which $\mathcal{M}_s$ has assigned high probability.

It is important to note that $\mathcal{M}_m$ and $\mathcal{M}_s$ operate at different temporal resolutions—note-level and semiquaver frame-level respectively—and that this difference becomes significant here. Suppose note ν is of duration $\Delta_\nu = \tau_\nu \delta$, where δ is the frame duration and τ_ν is the number of frames occupied by ν. Ideally, in order to get an accurate estimate of the joint probability of the note ν and the repeat π, one should consider the probability that $\mathcal{M}_s$ assigns to τ_ν *consecutive* frames of π. This would be expressed as

$$W_t(\nu) = P_t(\nu) \sum_{\pi \in \Pi_t^\nu} \prod_{k=0}^{\tau-1} \frac{Q_{t+k}(\pi)}{\mu_{t+k}^\pi}. \tag{3}$$

However, we found in our experiments that the single-step approximation (2) works well in practice and is less computationally intensive than (3).

Next, the weight distribution W_t is normalised to obtain a probability distribution R_t:

$$R_t(\nu) = \frac{W_t(\nu)}{\sum_{\nu \in N} W_t(\nu)}. \tag{4}$$

We may now sample a note ν_t from this distribution and update the internal state of the melodic model $\mathcal{M}_m$ with this observation.

It remains to update the state of the structure model $\mathcal{M}_s$ with some observed repeat. The note ν_t sampled at time t could be associated with any of the repeats that were consistent with it. We choose one by sampling π_t from a distribution S_t over $\Pi_t^{\nu_t}$ defined as

$$S_t(\pi) = \frac{Q_t(\pi)}{\sum_{\pi' \in \Pi_t^{\nu_t}} Q_t(\pi')}. \tag{5}$$

At this point the two models are misaligned due to the different time-scales they operate in, with $\mathcal{M}_m$ being τ semiquaver frames ahead of $\mathcal{M}_s$. Since each update of the state of $\mathcal{M}_s$ takes it ahead by just one semi-quaver frame, it is necessary to update $\mathcal{M}_s$ τ times repeatedly with the same structure vector so that it is once again aligned with $\mathcal{M}_m$.

At the end of the process described above, we have a melody note sampled from our melody model that has been influenced by StructureNet. StructureNet has also updated its own state according to the sampled note and is ready to influence the choice of next note.

5. EXPERIMENTS

We demonstrate the efficacy of StructureNet on a well-known dataset of melodies by comparing statistics over several musical quantities computed both on the dataset and compositions generated by the melody model alone and the melody and structure models combined. The results show that the presence of StructureNet leads to music that is more structured and closer in the statistics to the dataset. We also share the generated music to allow the reader to her or himself be the judge of our claims.

5.1 Dataset

We evaluate StructureNet on the cleaned Nottingham folk melody dataset that was released by the Jukedeck Research Team [23]. This publicly available dataset facilitates reproducibility. We carry out our experiments on the subset of 450 $4/4$ time-signature pieces out of the $1,548$ contained in it. Each piece of the dataset was truncated to its first 16 measures and transposed into the Key of C, and all upbeats at the beginning of each piece were removed prior to training. We used 20% (90 segments) of the data as the validation and the rest (360 segments) for training the models. StructureNet is also trained on the same dataset following the application of the repeat-tagging algorithm. However, one must note that this is not a requirement and a different dataset may be used for learning structure and could potentially lead to interesting results. StructureNet successfully induces structure in the generated melodies despite the few examples contained in the training data.

5.2 Training methodology

As mentioned earlier, both the structure network and the melody network are LSTMs and contain a single hidden layer. A Bayesian Optimisation based method was employed to carry out model selection. In the case of the melody model, the single best outcome of the grid search was used. As for StructureNet, ten models with the same best set of hyperparameters as determined by the model selection step, and different initial conditions, were trained and used in tandem with the melody model. This was done in order to be able to compute confidence intervals in the figures. The hidden layer size of each network was varied between 50 and 1000 in steps of 50 during model selection, which led to $n^s_{hid} = 950$ in the former and $n^m_{hid} = 250$ in the latter. Early stopping was used as a regulariser, such that the training was stopped and the best models thus far retrieved after no improvement in the validation cost for 25 epochs. The models were trained using the ADAM optimiser with an initial learning rate $\eta_{init} = 0.001$, and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

5.3 Evaluation

Our hypothesis is two-fold: (1) that repeat-related statistics computed over the melodies generated with StructureNet are closer to those over the dataset melodies than those over melodies generated without StructureNet, and (2) that non repeat-related statistics do not differ between the melodies generated by the melody model with and without StructureNet. This would demonstrate that the use of StructureNet leads to more structured melodies than are generated by the melody model on its own, and that are musically at least as similar to the original data as the melody model alone achieves. The statistics are:

1. **Repeat Count:** Number of repeats corresponding to various lookback values (in crotchets).
2. **Repeat Duration:** Number of repeats of various durations (in crotchets).
3. **Repeat Onsets:** Number of repeats beginning at various locations (in crotchets) in a piece.
4. **Pitch, start time and duration distributions:** Occurrence statistics of pitches, start times in measure, and durations.

The first three are repeat-related statistics and the rest are not. A histogram of each is first computed per collection of melodies (dataset, generations with and without StructureNet), and then normalised by the count of melodies in the collection to generate a probability distribution (as the counts vary between the different collections of melodies). The KL-Divergences (KLD) $\kappa_{data,SN}$ and $\kappa_{data,NoSN}$ between the distribution pairs (dataset, StructureNet) and (dataset, No StructureNet) respectively highlight the effect of introducing StructureNet (Table 1). Ideally, among the structure-related distributions, we would want $\kappa_{data,SN} < \kappa_{data,NoSN}$. And among the non-repeat-related distributions, we wish for $\kappa_{data,SN} \leq \kappa_{data,NoSN}$.

	$\kappa_{data,NoSN}$	$\kappa_{data,SN}$
Repeat Count (D)	0.0356 ± 0.0022	**0.0069 ± 0.0043**
Repeat Duration (D)	0.1071 ± 0.0047	**0.0389 ± 0.0168**
Repeat Onset (D)	0.0844 ± 0.0038	**0.0357 ± 0.0094**
Repeat Count (DI)	0.0511 ± 0.0049	**0.0173 ± 0.0095**
Repeat Duration (DI)	0.2402 ± 0.0069	**0.0634 ± 0.0352**
Repeat Onset (DI)	0.1209 ± 0.0073	**0.0639 ± 0.0194**
Repeat Count (all)	0.0483 ± 0.0035	**0.0083 ± 0.0045**
Repeat Duration (all)	0.0996 ± 0.0033	**0.025 ± 0.0081**
Repeat Onset (all)	0.0875 ± 0.0036	**0.031 ± 0.0103**
Pitch	0.0079 ± 0.0011	**0.0061 ± 0.0012**
Duration	0.0049 ± 0.0016	**0.0042 ± 0.0014**
Onset	0.058 ± 0.0081	**0.0275 ± 0.0082**

Table 1. KL-divergences between the training data and melodies generated with and without StructureNet (computed over 10 sets of 450 melodies generated with each trained StructureNet) for the Duration (D), Duration-Interval (DI) and all repeat types.

5.4 Observations

In Table 1, the KLD values show a greater match between the dataset and the set of generated melodies in the presence of StructureNet than in its absence. This holds true for both duration and duration-interval repeats. Figure 2 illustrates such similarities (over all repeat types) visually. One will see here that overall StructureNet (a) is conducive to the creation of longer repeats while generally having a positive effect on shorter ones as well, (b) is conducive to the creation of repeats that have lookback values similar to those in the dataset, particularly larger lookbacks (encouraging distant repeats), and (c) encourages repeats to begin on those metrical locations in a generated piece where they tend to occur in the dataset.

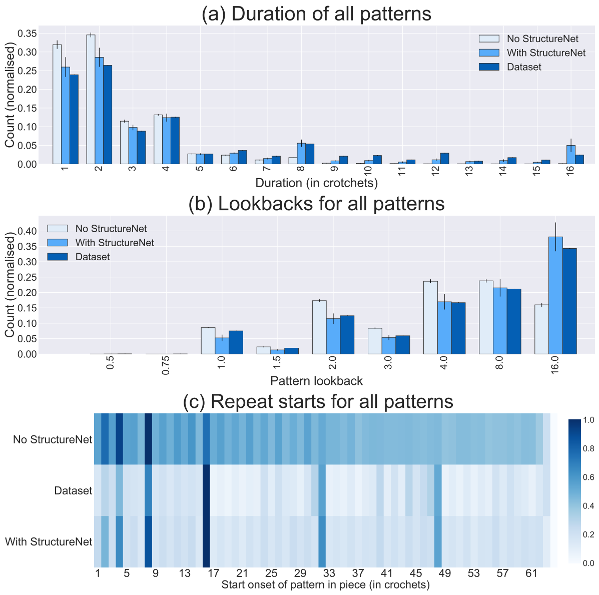

Figure 2. Repeat-related statistics of the dataset to the two generation modes (with/without StructureNet).

It is also evident from the set of three non-repeat-related statistics of Figure 3 that the presence of StructureNet has, more often than not, led to a better match of the generated melody statistics to the dataset. This is also supported by the very similar KLD values (often lower in the $\kappa_{data,NoSN}$ column) for these three musical quantities at the bottom of Table 1. And finally, each plot in Figure 4 shows the percentage of generated melodies with various degrees of free music in them. The three plots together reveal that using StructureNet reduces the proportion of free music (and thus increases the proportion of repeats) in the generated melodies in a way that more closely matches the proportions of free music and repeats in the dataset. Note that the statistics in Figures 2, 3 and 4 have been computed over the same number of melodies (of the same duration in measures). We have made a representative subset of melodies generated with and without StructureNet in the MIDI format available for scrutiny [3] .

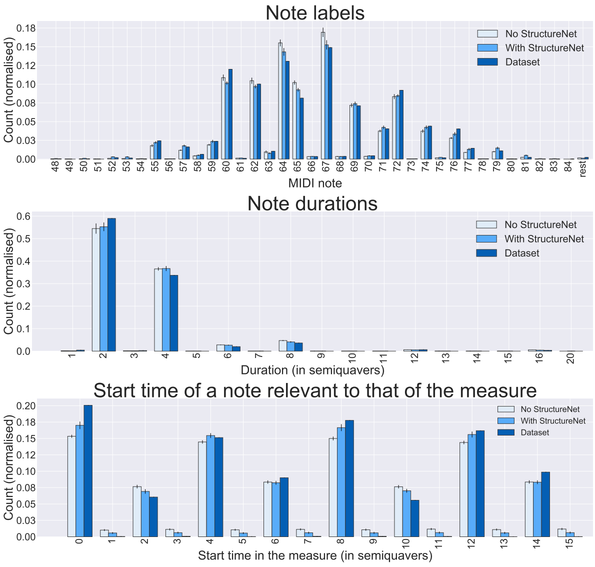

Figure 3. Non repeat-related statistics of the dataset to the two generation modes (with/without StructureNet).

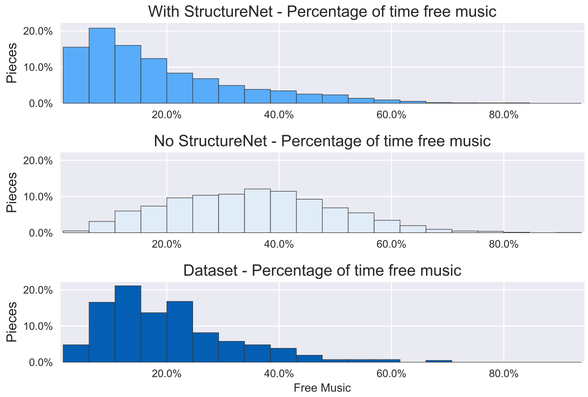

Figure 4. The amount of generated melodies with various degrees of free music in them.

6. CONCLUSIONS & FUTURE WORK

We introduced StructureNet - an RNN that influences the predictions of a melody model so as to give the generated melodies greater structure. We demonstrated using statistics, as well as several musical examples, that this model does indeed increase the probability of encountering longer and more distant (greater lookback) patterns in music generated by a melody model. Given these initially successful results, we foresee some interesting directions for future work. Firstly, we are interested in experimenting with a more evolved pattern detection algorithm such as SIATEC and COSIATEC [17]. This will lead to new feature representations over and beyond just repeats that can perhaps provide a better insight into musical structure to StructureNet. We would like to expand the three musical quantities introduced in Section 5.3 into a more comprehensive set of quantities that can lead to a more thorough evaluation of musical structure.

[3] https://goo.gl/hL9RhZ

7. REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *Intl. Conf. on Machine Learning*, pages 1881–1888. Omnipress, 2012.

[3] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep learning techniques for music generation-a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[4] Srikanth Cherla, Son N Tran, Artur d'Avila Garcez, and Tillman Weyde. Discriminative learning and inference in the recurrent temporal rbm for melody modelling. In *Intl. Joint Conf. on Neural Networks*, pages 1–8. IEEE, 2015.

[5] Joel E Cohen. Information theory and music. *Systems Research and Behavioral Science*, 7(2):137–163, 1962.

[6] Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[7] Nicholas Cook. *Music, imagination, and culture*. Oxford University Press, 1992.

[8] Jose David Fernndez and Francisco Vico. AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, 48:513–582, 2013.

[9] Jon Gillick, Kevin Tang, and Robert M Keller. Machine learning of jazz grammars. *Computer Music Journal*, 34(3):56–66, 2010.

[10] Kratarth Goel, Raunaq Vohra, and JK Sahoo. Polyphonic music generation by modeling temporal dependencies using a rnn-dbn. In *Intl. Conf. on Artificial Neural Networks*, pages 217–224. Springer, 2014.

[11] Dorien Herremans and Elaine Chew. Morpheus: generating structured music with constrained patterns and tension. *IEEE Trans. on Affective Computing*, 2017.

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

[13] Dominik Hörnel and Wolfram Menzel. Learning musical structure and style with neural networks. *Computer Music Journal*, 22(4):44–62, 1998.

[14] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints. *arXiv preprint arXiv:1612.04742*, 2016.

[15] Fred Lerdahl and Ray S Jackendoff. *A generative theory of tonal music*. MIT press, 1985.

[16] Qi Lyu, Zhiyong Wu, and Jun Zhu. Polyphonic music modelling with lstm-rtrbm. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 991–994. ACM, 2015.

[17] David Meredith. Cosiatec and siateccompress: Pattern discovery by geometric compression. In *Intl. Society for Music Information Retrieval Conf.* Intl. Society for Music Information Retrieval, 2013.

[18] Michael C Mozer. Connectionist music composition based on melodic, stylistic and psychophysical constraints. *Music and connectionism*, pages 195–211, 1991.

[19] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003.

[20] François Pachet and Pierre Roy. Markov constraints: steerable generation of markov sequences. *Constraints*, 16(2):148–172, 2011.

[21] Jean-Francois Paiement, Yves Grandvalet, and Samy Bengio. Predictive models for music. *Connection Science*, 21(2-3):253–272, 2009.

[22] Carles Roig, Lorenzo J Tardón, Isabel Barbancho, and Ana M Barbancho. Automatic melody composition based on a probabilistic model of music style and harmonic rules. *Knowledge-Based Systems*, 71:419–434, 2014.

[23] Jukedeck R&D Team. "Releasing a cleaned version of the Nottingham Dataset." Web blog post. *Jukedeck Research*, 7 Mar. 2017. Web. 30 Mar. 2018.

[24] Peter M Todd. A connectionist approach to algorithmic composition. *Computer Music Journal*, 13(4):27–43, 1989.

[25] Elliot Waite. "Generating Long-Term Structure in Songs and Stories." Web blog post. *Magenta*, 15 Jul. 2016. Web. 30 Mar. 2018.