HYBRID LONG- AND SHORT-TERM MODELS OF FOLK MELODIES

Srikanth Cherla[1,2] **Son N. Tran**[2] **Tillman Weyde**[1,2] **Artur d'Avila Garcez**[2]
[1]Music Informatics Research Group, Department of Computer Science, City University London
[2] Machine Learning Group, Department of Computer Science, City University London
{srikanth.cherla.1, son.tran.1, t.e.weyde, a.garcez}@city.ac.uk

ABSTRACT

In this paper, we present the results of a study on dynamic models for predicting sequences of musical pitch in melodies. Such models predict a probability distribution over the possible values of the next pitch in a sequence, which is obtained by combining the prediction of two components (1) a long-term model (LTM) learned offline on a corpus of melodies, as well as (2) a short-term model (STM) which incorporates context-specific information available during prediction. Both the LTM and the STM learn regularities in pitch sequences solely from data. The models are combined in an ensemble, wherein they are weighted by the relative entropies of their respective predictions. Going by previous work that demonstrates the success of Connectionist LTMs, we employ the recently proposed Recurrent Temporal Discriminative Restricted Boltzmann Machine (RTDRBM) as the LTM here. While it is indeed possible for the same model to also serve as an STM, our experiments showed that n-gram models tended to learn faster than the RTDRBM in an online setting and that the hybrid of an RTDRBM LTM and an n-gram STM gives us the best predictive performance yet on a corpus of monophonic chorale and folk melodies.

1. INTRODUCTION

In the present work, our interest is in learning a model to predict a probability distribution over the possible values of the pitch of a musical note in a melody given the sequence of notes leading up to it. The motivation for this stems from theoretical work in musicology and music cognition which attempts to explain various musical phenomena (such as style, genre and mood) in terms of patterns of fulfilment, prolongation and violation of musical expectation [10, 15, 19], i.e., that our perception of music is influenced by how its evolution in time conforms to, or deviates from our expectations. There exists empirical evidence suggesting that these expectations are shaped by an underlying mechanism of statistical learning [9], the consequences of which have also been observed in language [24]. This apparent commonality between the two domains has inspired the adoption of statistical models for word sequences in language and character sequences in text, to pitch sequences in melody [4, 6, 21, 31]. Previous work interpreting information theoretic concepts such as entropy and mutual information (which play a key role in language and text modelling) in the context of music [5, 16] contributed towards the adoption of these quantities in evaluating such *melody models*. Time-varying *entropy profiles* of predictions made by such models on musical pieces have been used for explaining stylistic implications of salient musical structures [7]. They have also been used to generate melodic stimuli in music cognition research [20]. Predictive models of music have also been used as Music Language Models in music transcription [26]. The reader is referred to [23] for a recent review on predictive machine learning models used in music research.

The melody models considered here contain two components - a long-term model (LTM), and a short-term model (STM) [6]. The parameters of each model are learned through exposure to appropriate data. From a machine learning perspective, the LTM is a model whose parameters are learned offline from a dataset of melodies. It represents more global stylistic characteristics acquired by a listener over a longer time-span. The parameters of the STM are learned online while making predictions on the test data, without any sequence learning occurring in it beforehand. The STM highlights the importance of context-specific information, available in a melody while it is being processed by the listener, in the generation of expectations. Predictions (in the form of probability distributions) made by each model about a certain musical event in a sequence are combined using ensemble methods, and this has been shown to improve the quality of predictions over individual models in the past [6,21]. The idea of combining corpus-based long-term and context-sensitive short-term predictions from different models was originally a feature of cache-based language models [12]. It was introduced in the context of music in [6], further extended in [21], and adopted in [7, 31].

To address the prediction task, we employ a recently proposed Connectionist model known as the Recurrent Temporal Discriminative Restricted Boltzmann Machine [3]. This model has been shown to have a predictive performance better than n-gram models and other standard Connectionist models on a corpus of monophonic melodies when used as an LTM. We begin by evaluating

© Srikanth Cherla, Son N. Tran, Tillman Weyde, Artur d'Avila Garcez. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Srikanth Cherla, Son N. Tran, Tillman Weyde, Artur d'Avila Garcez. "Hybrid Long- and Short-Term Models of Folk Melodies", 16th International Society for Music Information Retrieval Conference, 2015.

its utility as an STM by carrying out online learning in it, which has not been done previously. Experiments revealed that, while learning did indeed take place, it did not progress quickly enough (as a function of the number of data-points presented to the RTDRBM) to outperform existing state-of-the-art dynamic models based purely on n-grams [22]. On adopting the wisdom of previous work which demonstrated that n-gram models are indeed an effective choice as STMs, we found here that a hybrid prediction model which combines the predictions of an RTDRBM LTM and an n-gram STM achieves better predictive performance, and this also outperforms the state-of-the-art, purely n-gram based dynamic melody models on a corpus of 8 melody datasets. In this paper, we present the results of various LTM-STM combinations that we experimented with to arrive at this result and discuss our observations.

In the next section we formally introduce the task of melody modelling, and entropy-weighted combination strategies for LTMs and STMs. This is followed by a brief overview of the two types of prediction models involved in the present work, in Section 3. Various experiments in combining these models that led to the above mentioned optimal predictive performance are described in Section 4, followed by the conclusions in Section 5.

2. MELODY MODELLING

Our interest is in modelling musical pitch sequences through prediction. The task of music prediction addressed here has strong parallels with previous work in language modelling [14]. Thus, the analogy to natural language is used here to explain it. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of subsequences of words occurring in a language L. A statistical language model (SLM) can be represented by the conditional probability of the next word $w^{(T)}$ given all the previous ones $[w^{(1)}, \ldots, w^{(T-1)}]$ (written $w^{(1:T-1)}$), as

$$P(w^{(1:T)}) = \prod_{t=1}^{T} P(w^{(t)}|w^{(1:t-1)}) \,. \qquad (1)$$

The present work treats notes in a monophonic melody analogous to words in the above language example. This is inspired by [6] where a similar analogy was made between sequences of characters in the English language and notes in music. We use an event-based representation of music, where the occurrence of each note is treated as a *musical event*. Much in the same way as an SLM, a system for music prediction models the conditional distribution $P(s^{(t)}|s^{(1:t-1)})$ given a sequence $s^{(1:T)}$ of musical events [4, 6, 22] from a musical language S, such that $s^{(t)} \in [S]$, where $[S]$ is the set of symbols (musical pitch values) in S. For each prediction, context information is obtained from the events $s^{(1:t-1)}$ preceding $s^{(t)}$. Although a range of musical features (such as musical pitch, note duration, inter-onset interval, etc.) may be extracted from each musical event as explained in [6], we limit our attention to sequences of musical pitch. And the symbols that make up these sequences are MIDI values of the pitches which occur in a particular dataset.

2.1 Long- and Short-term Models

In the present work, we make a distinction between two types of prediction models, as introduced previously in the context of *Multiple Viewpoints for Music Prediction* [6]. The first is known as a Long-Term Model (LTM). This model is learned offline on a corpus of melodies (training data), its parameters thus being finalized beforehand and kept constant during the prediction stage. It represents more global stylistic characteristics acquired by a listener over a longer time-span. And the second is what is known as the Short-Term Model (STM). It highlights the importance of context-specific information, available in a melody while it is being processed by the listener, in the generation of expectations. The distinction between the long- and short-term models is also akin to the that made in [11] between "schematic" (LTM) and "veridical" (STM) knowledge in a modular view on music processing. A variant of the LTM which is also considered here was introduced in [22]. This is the LTM+, and in addition to being learned offline on a corpus of melodies like the LTM, it is also updated while making predictions just like the STM. Another distinction between the LTM+ and the STM is that the former is continuously updated across melodies, while the latter is re-initialized after each melody in the test set.

2.2 Combining the LTM & STM

It was demonstrated in [6, 21] that an entropy-weighted combination of the predictions of two or more n-gram models typically results in ensembles with better predictive performance than any of the individual models. As it is the predicted distributions which are combined, this approach is independent of the types of models involved. Here, we briefly describe two rules for creating such ensembles. Let M be a set of models and $P_m(s)$ be the probability assigned to symbol $s \in [S]$ by model m. The first involves taking a weighted arithmetic mean of their respective predictions. This is the *Mean* combination rule, defined as

$$P(s) = \frac{\sum_{m \in M} w_m P_m(s)}{\sum_{m \in M} w_m} \qquad (2)$$

where each of the weights w_m depends on the entropy of the distribution generated by the corresponding model m in the combination such that greater entropy (and hence uncertainty) is associated with a lower weight [6]. The weights are given by the expression $w_m = H_{rel}(P_m)^{-b}$, where the relative entropy $H_{rel}(P_m)$ is

$$H_{rel}(P_m) = \begin{cases} H(P_m)/H_{max}(P_m), & \text{if } H_{max}([S]) > 0 \\ 1, & \text{otherwise} \end{cases} \qquad (3)$$

The best value of the combination bias $b \geq 0$ is determined through cross-validation. When $b = 0$, all the combined models have the same weight. The quantities H and H_{max}

are respectively the entropy of the prediction and the maximum entropy of predictions over the symbol space $[S]$, and are defined as

$$H(P) = -\sum_{s\in[S]} P(s)\log_2 P(s)\ . \tag{4}$$

$$H_{max}(P) = \log_2 |S|.$$

where $P(X = s)$ is the probability mass function of a random variable X distributed over the discrete alphabet $[S]$ such that the individual probabilities are independent and sum to 1.

The second — the *Product* combination rule, is computed similarly as the weighted geometric mean of the probability distributions. This is given by

$$P(s) = \frac{1}{R}\left(\prod_{m\in M} P_m(s)^{w_m}\right)^{\frac{1}{\sum_{m\in M} w_m}} \tag{5}$$

where R is a normalisation constant which ensures that the resulting distribution over S sums to unity. The weights w_m in this case are obtained in the same manner as in the case of the Mean combination rule. It was observed in a previous application of these two combination methods to melody modelling [21], that the Product rule resulted in a greater improvement in predictive performance.

3. PREDICTION MODELS

Before moving on to the experiments carried out on different LTM-STM combinations in the next section, here we provide a quick overview of the two classes of prediction models that have been employed for this purpose. The first is the Recurrent Temporal Discriminative Restricted Boltzmann Machine, and the other is the n-gram Model.

3.1 Recurrent Temporal Discriminative RBM

The Recurrent Temporal Discriminative Restricted Boltzmann Machine (RTDRBM) [3] was proposed by the authors as the discriminative equivalent of the Recurrent Temporal Restricted Boltzmann Machine (RTRBM) [28]. Both models are identical in structure, and are composed of a sequence of Restricted Boltzmann Machines (RBM) [27], where the visible and hidden layers of the RBM at time-step t are conditioned on the mean-field values of the hidden layer of that at $(t-1)$ through a set of time-dependent model parameters. The RTDRBM learns the distribution $P(\mathbf{y}^{(1:T)}|\mathbf{x}^{(1:T)})$ over a sequence of input-label pairs $\{\mathbf{x}^{(1:T)}, \mathbf{y}^{(1:T)}\}$, in contrast to the RTRBM which learns the joint probability of the entire sequence $P(\mathbf{y}^{(1:T)}, \mathbf{x}^{(1:T)})$ [1].

The RTDRBM (Figure 1) is obtained by carrying out discriminative learning and inference as put forward in the Discriminative RBM (DRBM) [13], in a temporal setting by incorporating the recurrent structure of the RTRBM which was originally proposed as a generative model for high-dimensional sequences. This results in the following expression for the posterior probabilities at time-step t:

$$P(\mathbf{y}^{(t)}|\mathbf{x}^{(1:t)}) = P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, \hat{\mathbf{h}}^{(t-1)}) \tag{6}$$

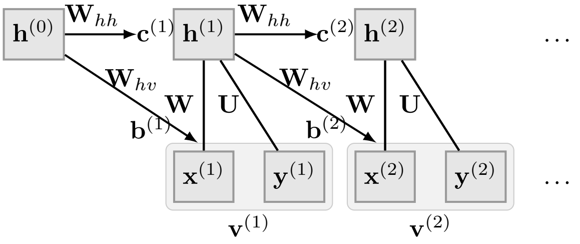

Figure 1: The architecture of the RTDRBM, in which the biases of the visible and hidden layers $\mathbf{b}^{(t)}$ and $\mathbf{c}^{(t)}$ respectively at time-step t are conditioned on the mean-field values of the hidden layer of the RBM $\hat{\mathbf{h}}^{(t-1)}$ at time-step $(t-1)$. This is also a feature of the RTRBM.

It takes into account temporal information carried forward from the previous time-step through the mean-field values of the hidden units $\hat{\mathbf{h}}^{(t-1)}$ [3]. This can be extended to an entire sequence of T events as follows:

$$P(\mathbf{y}^{(1:T)}|\mathbf{x}^{(1:T)}) = \prod_{t=1}^{T} p(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, \hat{\mathbf{h}}^{(t-1)}) \tag{7}$$

One can thus learn the model by maximizing the log-likelihood function:

$$\begin{aligned}\mathcal{O} &= \log P(\mathbf{y}^{(1:T)}|\mathbf{x}^{(1:T)}) \\ &= \sum_{t=1}^{T} \log P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, \hat{\mathbf{h}}^{(t-1)})\ .\end{aligned} \tag{8}$$

Learning here involves updating the model's parameters as dictated by the Backpropagation Through Time (BPTT) algorithm [30]. It was demonstrated in [3] that the RTDRBM outperformed the RTRBM, n-grams and a set of standard Connectionist models on a corpus of 8 different datasets of chorale and folk melodies of varying sizes and complexities when learned offline. In the pitch prediction task of Section 2, the one-hot encoding of the musical event $s^{(t)}$ (which is to be predicted) substitutes the label $\mathbf{y}^{(t)}$ in (6), whereas that of the most recent event from the context $s^{(t-1)}$ substitutes the input $\mathbf{x}^{(t)}$.

3.2 n-gram Model

The n-gram model is a statistical model of sequences that relies on the simplifying assumption that the probability of an event (or in the present case, a musical event) in a sequence depends only on the $(n-1)$ immediately preceding events [14]. This is known as the Markov assumption, and is applied to model an event sequence $s^{(1:T)}$ as

$$P(s^{(1:T)}) = \prod_{t=1}^{T} P(s^{(t)}|s^{(t-n+1:t-1)})\ . \tag{9}$$

where n is known as the order of the n-gram. The model can be represented by a state transition graph, or by a *transition matrix*. Maximum-Likelihood Estimation can be carried out to estimate the parameters of the n-gram model (its transition probabilities) as

$$P(s^{(t)}|s^{(t-n+1:t-1)}) = \frac{N(s^{(t-n+1:t)})}{N(s^{(t-n+1:t-1)})} \tag{10}$$

where $N(s^{(t_1:t_2)})$ is the number of occurrences of a sequence $s^{(t_1:t_2)}$ in the data. As we shall see in Section 4, this simple learning rule is advantageous in an online-learning scenario where the model needs to be constantly updated as it encounters new data. As n-grams rely explicitly on the occurrence frequencies of sequences, it is often the case that the model comes across a never-before-encountered context on which to predict the future event, and this is more common in higher order models. This issue has been dealt with by using *smoothed n-grams* [2] that use lower-order transition probabilities for generating approximations (through interpolation with or scaling of) higher-order probabilities. This also applicable to events that lack a valid context, i.e. $\{s^{(t)} \mid 1 \leq t \leq (n-1)\}$.

The present work employs two of the best variants of the n-gram model evaluated for melody modelling in [22] exclusively as STMs, as an alternative to the RTDRBM which performs poorly in this role (Table 3). Both variants are of unbounded order, wherein they take into account the longest available matching context (of immediately preceding musical events) in order to make a prediction. The first of these (referred to as C^*I) uses the interpolated smoothing method proposed in [18] to account for unfamiliar contexts. The second (referred to as X^*UI) uses a Poisson process based interpolated smoothing method [18] with update exclusion [17]. We refer the interested reader to [22] for further details on these two models.

4. EXPERIMENTAL RESULTS

We evaluate six different LTM-STM combinations. These are listed in Table 1. Also, C^*I and X^*UI are the names of the two best STMs evaluated in a previous study of n-gram based melody models [22].

(a) **LTM:** RTDRBM	**STM:** n-gram (X*UI)
(b) **LTM:** RTDRBM	**STM:** n-gram (C*I)
(c) **LTM:** RTDRBM	**STM:** RTDRBM
(d) **LTM+:** RTDRBM	**STM:** n-gram (X*UI)
(e) **LTM+:** RTDRBM	**STM:** n-gram (C*I)
(f) **LTM+:** RTDRBM	**STM:** RTDRBM

Table 1: Various LTM-STM combinations evaluated here.

Each of the combined models was evaluated on 8 melody datasets of different sizes and styles. Prediction cross-entropy was used as the evaluation measure. It was found that combination (b) had the best predictive performance. Furthermore, each case involving an LTM was better than its LTM+ counterpart. And finally, the n-grams consistently proved to be a better choice than the RTDRBM as STMs when combined with the same LTM.

Dataset	No. events	$\|X\|$
Yugoslavian folk songs	2691	25
Alsatian folk songs	4496	32
Swiss folk songs	4586	34
Austrian folk songs	5306	35
German folk songs	8393	27
Canadian folk songs	8553	25
Chorale melodies	9227	21
Chinese folk songs	11056	41

Table 2: Melody datasets used for evaluation with their respective total number of musical events and number of prediction categories.

4.1 Data

Evaluation was carried out on a corpus of 8 datasets of monophonic MIDI melodies from the Essen Folk Song Collection [1] [25]. The corpus covers a range of musical styles and was previously used in [4, 22] to evaluate their respective prediction models. It contains folk melodies of 7 different traditions, and chorale melodies (Table 2). All melodies are encoded in the ***kern* format in each dataset, and were parsed using the *Music21* Python library [8]. Musical pitch, which occurs as sequences of integer values, is treated as a discrete random variable X, which can assume any of $|X|$ distinct values (or prediction categories).

4.2 Evaluation Measure

Given that the models predict a probability distribution over X at every time-step, their goal may be viewed as one of minimizing the distance between this predicted distribution and that representing the correct class label (the value of the next pitch). An obvious choice of evaluation measure in this case would be the information theoretic quantity which calculates this distance: relative entropy. Here we use a measure derived from it known as cross-entropy (H_c), in order to compare our results with previous work [22]. This gives us the mean divergence between the entropy calculated from the predicted distribution and that of the correct prediction label (and can be interpreted as the distance between these two distributions) for every sample in some given data. It can be computed over all the events belonging to different sequences in the test data $\mathcal{D}_{test}$, as

$$H_c(P_{mod}, \mathcal{D}_{test}) = \frac{-\sum_{s \in \mathcal{D}_{test}} \sum_{t=1}^{T_s} \log_2 P_{mod}(s^{(t)}|s^{(1:t-1)})}{\sum_{s \in \mathcal{D}_{test}} T_s} \tag{11}$$

[1] Website: http://kern.ccarh.org/browse?l=essen

where P_{mod} is the probability assigned by the model to the pitch of the event $s^{(t)}$ in the melody $s \in \mathcal{D}_{test}$ given its preceding context, and T_s is the length of s. Cross-entropy approaches the true entropy as the number of test samples, i.e., the denominator in (11) increases.

4.3 Methodology

The models are evaluated using 10-fold cross-validation. We use randomised folds identical to those used in previous work [4, 22] to facilitate fair comparison[2]. A small part of the training set (5%) in each fold is extracted as the validation set for model selection over the various hyperparameters described below. This procedure is repeated independently for each of the 8 datasets in the corpus.

The RTDRBM LTMs were learned (offline) up to a maximum of 250 epochs using mini-batch gradient descent on the training set, and that with the best validation set score was chosen for evaluation on the test set. A grid search was carried out to determine the best set of hyperparameters for each model. These constitute the learning rate η, the L_1 and L_2 regularization (λ_1 and λ_2 respectively) and the number of hidden units n_{hid}. For each of the models, η was varied as $\{0.01, 0.05\}$, and n_{hid} as $\{10, 25, 50, 100, 200\}$. Both L_1 and L_2 decay were set to identical values $\lambda_1 = \lambda_2 = \lambda$ which was either on ($\lambda = 0.0001$) or off ($\lambda = 0.0000$). Learning rate was made to decay according to the schedule $\eta_t = \eta_{init}/(1 + t/\tau)$, where $\tau = 50$.

The RTDRBM LTM+s and STMs were learned (online) using stochastic gradient descent, where model parameters were updated after each time-step during prediction on the test set, with the only distinction between the two being that the parameters of the former are initialized to those of the best LTM learned offline on the dataset. As explained in Section 2.1, the LTM+ is continuously updated across melodies, while the STM is re-initialized after each melody in the test set. Since each of the STMs is expected to learn a smaller number of patterns than its corresponding LTM, we decided to extend the model selection with much smaller models as well ($n_{hid} \in \{2, 5, 10, 20, 100, 200\}$), with the remaining hyperparameters kept the same, and a constant learning rate i.e., $\eta_t = \eta_{init} = 0.01$.

The combination bias parameter b for computing the entropy-based weights w_m was varied as $b = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 16, 32\}$, as in [21]. This range was used for both combination rules, following the example of [21].

4.4 Results & Discussion

Table 3 shows the predictive performance of various LTM-STM combination rules evaluated here together with the corresponding combination bias value used, averaged across all 8 datasets. The bottom row of this table corresponds to the performance of the purely n-gram based melody model in [22], which we compare the models evaluated here with.

Model	LTM	STM	Mix.	b_m	Prod.	b_p
(a)	2.712	3.053	2.480	3	2.496	1
(b)	2.712	3.046	**2.421**	**4**	**2.487**	**1**
(c)	2.712	3.363	2.674	5	2.703	1
(d)	2.756	3.053	2.574	2	2.563	1
(e)	2.756	3.046	2.540	2	2.581	1
(f)	2.756	3.363	2.749	5	2.773	1
n-gram	2.614	3.147	2.479	2	N/A	N/A

Table 3: Predictive performance of various model combinations listed in Table 1, in comparison with a purely n-gram based melody model (bottom row). Each row of the table contains the prediction cross-entropies of the constituent LTM (or LTM+), STM, and the combination of these two using the Mean and Product rules together with the respective biases. A lower value of cross-entropy reflects more accurate predictions.

In each case, the RTDRBM LTM has 100 hidden units (found to be the best in the model selection procedure). Despite the extended grid search for the STMs, it was found that the optimal number of hidden units was 100 in that case as well.

The first thing to note is that combining the models (using either of the two combination rules) results in an improvement in predictive performance over each of the constituent models. Furthermore, the Mean combination rule results in slightly better prediction cross-entropies than Product rule. This can be explained by considering the basic properties of the two rules, as concluded by a previous study comparing them [29]. The Mean combination rule is useful in case of identical or very highly correlated feature spaces (which holds true in the present case) in which classifiers make independent errors. Furthermore, this rule is generally more fault tolerant in the case of poor posterior probability estimates (which is indeed the case here with the STM being learned afresh at the start of each melody), whereas the Product rule emphasizes the points of agreement between the two models and is apt where classifiers make small estimation errors. The best combined model (RTDRBM LTM; n-gram ($C^{*}I$) STM) performs slightly better than the best purely n-gram based melody model in [22]. In the case of both the Mean and Product rules, it was found that smaller values of the combination bias parameter were preferred over larger ones, with a value of 1 being consistently optimal in the case of the latter.

Another observation is regarding the LTM and LTM+, where the latter performs slightly worse when compared to the former. This contrasts what has been previously observed when using n-gram models, where there was an improvement from the LTM to the LTM+ [22]. One possible reason for this could be the absence of any new sequential regularities in the test data to update the already optimized LTM with, since both the training and test sequences have been sampled from the same data distribution. Alternatively, the gradient-based optimization procedure employed here for online learning (stochastic

[2] Information about the training/test split in the 10 folds was obtained from the authors of [22]

gradient-descent) might not be the ideal choice for updating the model quickly enough to facilitate an improvement in the predictions. The latter reason could also explain the relatively poor performance of the RTDRBM STMs when compared to the STMs based on n-grams. This requires further investigation.

5. CONCLUSIONS & FUTURE WORK

This paper presented a study on models for melody prediction with a long-term and a short-term component (LTM and STM respectively). While all the LTMs explored here are based on the Recurrent Temporal Discriminative RBM (RTDRBM), the STMs are based both on the RTDRBM and n-gram models. It was found that, while the RTDRBMs are indeed a suitable choice when learned offline as LTMs [3], they fail to achieve a predictive performance as good as that of the n-gram models considered here in an online setting (as in the case of the LTM+ and the STM). The best model in the present work is a combination of an RTDRBM LTM and an n-gram STM which performs better than the state-of-the-art model based purely on n-grams. Among the two combination rules - Mean and Product - it was found that the former rule works better with the models and data used here.

One issue that remains unresolved in the present work, and requires investigation in the future, is the lack of improvement in predictions during online learning in the RTDRBM LTM. Another extension to the models employed here is to incorporate additional melodic features as inputs, as detailed in *Multiple Viewpoints for Music Prediction* [6], and to examine how this would improve or worsen the predictive performance over the existing models. And finally, previous work with LTMs and STMs based purely on n-gram models has found the predictions made by these models to reflect the musical expectations of human subjects. This is also relevant to the models explored here, and is of interest in the future.

6. ACKNOWLEDGEMENTS

Srikanth Cherla and Son N. Tran are supported by Ph.D. studentships from City University London. The authors would like to thank Marcus Pearce (Queen Mary University of London) and Senanayak Sesh Kumar Karri (INRIA Rocqencourt) for their advice and helpful discussions related to parts of this paper.

7. REFERENCES

[1] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. In *International Conference on Machine Learning*, pages 1159–1166, 2012.

[2] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.

[3] Srikanth Cherla, Son Tran, Artur d'Avila Garcez, and Tillman Weyde. Discriminative Learning and Inference in the Recurrent Temporal RBM for Melody Modelling. In *International Joint Conference on Neural Networks*, 2015.

[4] Srikanth Cherla, Tillman Weyde, Artur d'Avila Garcez, and Marcus Pearce. A Distributed Model for Multiple-Viewpoint Melodic Prediction. In *International Society for Music Information Retrieval Conference*, pages 15–20, 2013.

[5] Joel E Cohen. Information Theory and Music. *Behavioral Science*, 7(2):137–163, 1962.

[6] Darrell Conklin and Ian Witten. Multiple Viewpoint Systems for Music Prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[7] Greg Cox. On the Relationship Between Entropy and Meaning in Music: An Exploration with Recurrent Neural Networks. In *Annual Conference of the Cognitive Science Society*, pages 429–434, 2010.

[8] Michael Cuthbert and Christopher Ariza. music21: A Toolkit for Computer-aided Musicology and Symbolic Music Data. In *International Society for Music Information Retrieval Conference*, pages 637–642, 2010.

[9] Tuomas Eerola, Petri Toiviainen, and Carol Krumhansl. Real-Time Prediction of Melodies: Continuous Predictability Judgements and Dynamic Models. In *International Conference on Music Perception and Cognition*, pages 473–476, 2002.

[10] David Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, 2006.

[11] Timothy Justus and Jamshed Bharucha. Modularity in Musical Processing: The Automaticity of Harmonic Priming. *Journal of Experimental Psychology: Human Perception and Performance*, 27(4):1000–1011, 2001.

[12] Roland Kuhn and Renato De Mori. A Cache-based Natural Language Model for Speech Recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(6):570–583, 1990.

[13] Hugo Larochelle and Yoshua Bengio. Classification using Discriminative Restricted Boltzmann Machines. In *International Conference on Machine Learning*, pages 536–543, 2008.

[14] Christopher Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[15] Leonard Meyer. *Emotion and Meaning in Music*. University of Chicago Press, 1956.

[16] Leonard Meyer. Meaning in Music and Information Theory. *Journal of Aesthetics and Art Criticism*, pages 412–424, 1957.

[17] Alistair Moffat. Implementing the PPM data compression scheme. *Communications, IEEE Transactions on*, 38(11):1917–1921, 1990.

[18] Alistair Moffat, Radford Neal, and Ian Witten. Arithmetic Coding Revisited. *ACM Transactions on Information Systems (TOIS)*, 16(3):256–294, 1998.

[19] Eugene Narmour. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. University of Chicago Press, 1992.

[20] Diana Omigie, Marcus Pearce, Victoria Williamson, and Lauren Stewart. Electrophysiological Correlates of Melodic Processing in Congenital Amusia. *Neuropsychologia*, 2013.

[21] Marcus Pearce. *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD thesis, City University London, 2005.

[22] Marcus Pearce and Geraint Wiggins. Improved Methods for Statistical Modelling of Monophonic Music. *Journal of New Music Research*, 33(4):367–385, 2004.

[23] Martin Rohrmeier and Stefan Koelsch. Predictive Information Processing in Music Cognition. A Critical Review. *International Journal of Psychophysiology*, 83(2):164–175, 2012.

[24] Jenny Saffran, Elizabeth Johnson, Richard Aslin, and Elissa Newport. Statistical Learning of Tone Sequences by Human Infants and Adults. *Cognition*, 70(1):27–52, 1999.

[25] Helmut Schaffrath and David Huron. The Essen Folksong Collection in the Humdrum Kern Format. 1995.

[26] Siddharth Sigtia, Emmanouil Benetos, Nicolas Boulanger-Lewandowski, Tillman Weyde, Artur d'Avila Garcez, and Simon Dixon. A Hybrid Recurrent Neural Network For Music Transcription. In *International Conference on Acoustics Speech and Signal Processing*, 2015.

[27] Paul Smolensky. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, pages 194–281. MIT Press, 1986.

[28] Ilya Sutskever, Geoffrey Hinton, and Graham Taylor. The Recurrent Temporal Restricted Boltzmann Machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009.

[29] David Tax, Martijn Van Breukelen, Robert Duin, and Josef Kittler. Combining Multiple Classifiers by Averaging or by Multiplying? *Pattern Recognition*, 33(9):1475–1485, 2000.

[30] Paul Werbos. Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[31] Raymond Whorley. *The Construction and Evaluation of Statistical Models of Melody and Harmony*. PhD thesis, Goldsmiths, University of London, 2013.