A Model Comparison for Chord Prediction on the Annotated Beethoven Corpus

Kristoffer Landsnes
EPFL
kristoffer.landsnes@epfl.ch

Liana Mehrabyan
EPFL
liana.mehrabyan@epfl.ch

Victor Wiklund
EPFL
victor.wiklund@epfl.ch

Robert Lieck
EPFL
robert.lieck@epfl.ch

Fabian C. Moss
EPFL
fabian.moss@epfl.ch

Martin Rohrmeier
EPFL
martin.rohrmeier@epfl.ch

ABSTRACT

This paper models predictive processing of chords using a corpus of Ludwig van Beethoven's string quartets. A recently published dataset consisting of expert harmonic analyses of all Beethoven string quartets was used to evaluate an n-gram language model as well as a recurrent neural network (RNN) architecture based on long-short-term memory (LSTM). We compare model performances over different periods of Beethoven's creative activity and provide a baseline for future research on predictive processing of chords in full Roman numeral representation on this dataset.

1. INTRODUCTION

Predictive processing and the formation of expectancies are core capacities of human cognition that also play a fundamental role in music perception and cognition [1–4]. Musical expectancies are essential for processes at different time-scales, such as for musical interaction and synchronization, as well as for musical tension and the play with emotional effects [5, 6]. Musical expectancy has also been understood to be culture- and style-dependent and to be grounded in musical knowledge that is acquired through processes of implicit or statistical learning [1, 7, 8]. The modelling of predictive processing and the formation of expectancies is thus of core importance for computational models of music and requires a learning-based approach.

Musical expectancy has been studied in terms of melody, harmony and rhythm, where the task is to predict the next note, chord, onset or a combination thereof. In the general case of polyphonic music, it is a non-trivial problem to find a consistent representation of musical content and to accurately define what events should be predicted. Many past approaches have, therefore, simplified the problem to predicting a single stream of events from a fixed alphabet, such as melodic notes or chord events. This task is structurally closely related to modelling natural language, and similar approaches have been taken in both fields. Most notably, one can distinguish models that use a finite-length context, such as n-gram or kth-order Markov models, from models that use a latent state to capture longer dependencies, such as hidden Markov models (HMMs) [9] and recurrent neural networks (RNNs) [10, 11].

In this paper, we focus on modeling the prediction of a chord symbol given a harmonic context based on a recent data set comprising expert annotations of the 16 Beethoven string quartets [12], subsumed under nine different opus numbers which formed the basic grouping for all analyses. To this end, we evaluate a standard n-gram model as well as a state-of-the-art RNN architecture based on long short-term memory (LSTM) [13]. We report and compare accuracy results of the two models over different opera and discuss our results form a technical as well as from a music theoretical point of view.

2. METHODS

2.1 Data and Preprocessing

The data used for this project contain the expert harmonic analyses of all 16 Beethoven string quartets incorporated in nine opera: Op. 18 (6 quartets), op. 95 (3 quartets) and 7 other opera, each containing one quartet. We group the string quartets by opus number assuming that an opus constitutes a coherent unit of a musical work with pieces that are not independent of each other and should thus be treated as dependent data in the training procedure. Features in the data include global and local keys, beat, time signature, opus and movement numbers. The chord annotation format used in the dataset is a formalised version of Roman numeral notation, the most common music theoretic set of symbols for harmonic analysis. In addition to the key, the scale degree, and the figured bass, the chord annotations include information on suspensions, added notes and pedal notes. Table 1 demonstrates several examples of of this annotation format. A more detailed explanation can be found at the official documentation of the data [12]. This annotation format is much richer than what is commonly found in harmonic corpora and thus implies a particularly challenging learning problem.

A total number of $28,095$ chord labels are annotated resulting in $1,730$ unique items. More than 1,500 chords occur less than 10 times throughout the whole corpus of 16 quartets (908 of which occur only once), while the top 5 chords occur more than 1,000 times throughout all quartets. This distribution is similar to the Zipf distributions

Notation	Interpretation
V43	a dominant seventh chord in second inversion
ii%7	a half-diminished seventh chord on the second scale degree
IV6//	a major triad on the fourth scale degree in its first inversion at a phrase end
vi(+9)	a minor triad on the sixth scale degree with an added ninth
V[V	a dominant triad over the pedal tone on fifth scale degree

Table 1. Examples for chord symbols in the dataset and interpretations.

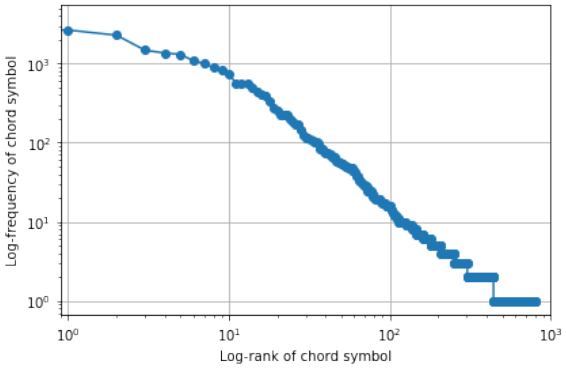

Figure 1. Frequency of chord symbols on log-log scale.

frequently found in natural language and music [14, 15].

In order to reduce the absolute number of chord classes, several preprocessing rules were established. Chord symbols on top of a pedal, e.g. a suspended tone in the Cello, were disregarded because their harmonic function is more ambiguous. As a result, the number of chord categories was drastically reduced to only 800. Figure 1 represents the resulting distribution of chord ranks vs. chord frequencies after preprocessing.

2.2 N-gram Language Model

Our goal is the prediction of chord symbols, given some harmonic context. The simplest choice for a baseline model is to use an n-gram language model, which estimates the probability of the i^{th} word w_i based on the context of the previous $n-1$ words $w_{i-(n-1)} \dots w_{i-1}$as follows:

$$P(w_i|w_{i-(n-1)} \dots w_{i-1}) = \frac{C(w_{i-(n-1)} \dots w_i)}{C(w_{i-(n-1)} \dots w_{i-1})}\,, \tag{1}$$

where $C(\cdot)$ counts the number of times the respective sequence of words occurs in the training data. In order to find the optimal n-gram length, hyperparameter tuning was performed. Values of $n = 2, 3, \dots, 10$ were used to evaluate results by cross validation: for each iteration, the model was trained on the whole corpus except one opus, which was reserved for validation purposes. As a simple n-gram model such as (1) can not handle unseen events, we use add-one smoothing [16] by adding one prior count to all symbols and adjusting the denominator of (1) accordingly

$$P(w_i|w_{i-(n-1)} \dots w_{i-1}) = \frac{C(w_{i-(n-1)} \dots w_i) + 1}{C(w_{i-(n-1)} \dots w_{i-1}) + V}\,, \tag{2}$$

where V is the total number of unique chords in the corpus.

Parameter	**Values**
Sequence Lengths [chords]	[10, 20, 40, 80, 160]
Amount of layers	[1, 2, 3, 4, 5]
Layer type	[LSTM, Bi-directional LSTM]
Amount of neurons	[8, 16, 32, 64, 128, 256, 512]
Dropout strength	[0, .1, .2, .3, .4, .5]
L2-regularization	[0, .001, .005, .01, .05, .1, .5]

Table 2. Model parameters explored

Layer	**Description**
LSTM	256 neurons, return sequences = True, L2 = 0
Dropout	Strength = 0.3
LSTM	64 neurons, return sequences = False, L2 = 0
Dropout	Strength = 0.3
Dense	821 neurons, activation = sigmoid, L2 = 0

Table 3. Model layout

2.3 Neural Network

As a more complex model for the prediction of chord symbols we used a Recurrent Neural Network (RNN) with Long Short-Term Memory cells (LSTM). This model was selected because this type of network has shown promise in sequence prediction tasks with long term dependencies [17] and thus seems suitable for an application to music. Moreover, it allows us to compare the more complex RNN model with the more basic n-gram model.

The design of the model architecture was based on related work [18, 19] after which modifications were tested manually by maximizing for validation accuracy on 10% of the data while training on the remaining 90%. Different configurations of sequence lengths, dropout, amount of layers, type of layers, and amount of neurons were tested. Tuning of the L_2-regularisation strength and dropout rate was then done with a nine-fold cross validation using the distinct opera as cross-validation folds. We tested parameters in the ranges shown in Table 2. Our final network architecture is shown in Table 3.

We also tested replacing the initial LSTM layer with a convolutional layer and performed a grid search over kernel size and amount of filters. While the training phase was notably faster, peak validation accuracy was a bit lower than our final final LSTM architecture.

For the activation functions we used the defaults provided with the Keras library [20], which is $\tanh$. To normalize the network output to a categorical distribution we used a

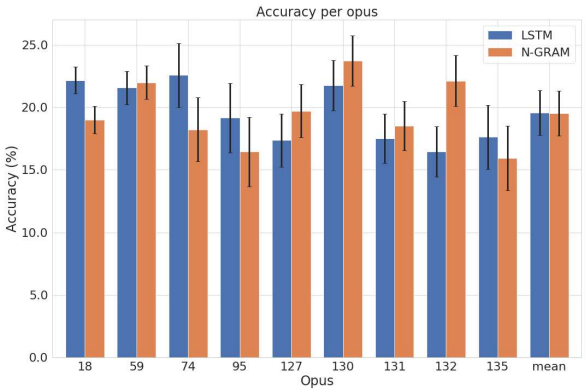

Figure 2. Comparative accuracy of a n-gram and LSTM model using nine-fold cross validation. The error bars are defined as as 1/sqrt(n), n being the length of the opus

softmax function

$$S(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}. \quad (3)$$

For training we used the Adaptive Moment Estimation (ADAM) optimizer [21], which has proven to yield robust performance based on prior work in the field of neural networks and deep learning, primarily ascribed to the adaptivity of the learning rate it employs.

opus	LSTM	N-GRAM
18	0.2217	0.1900
59	0.2157	0.2200
74	0.2257	0.1823
95	0.1917	0.1645
127	0.1738	0.1972
130	0.2175	0.2373
131	0.1750	0.1852
132	0.1646	0.2212
135	0.1763	0.1594
mean	0.1958	0.1952

Table 4. Results

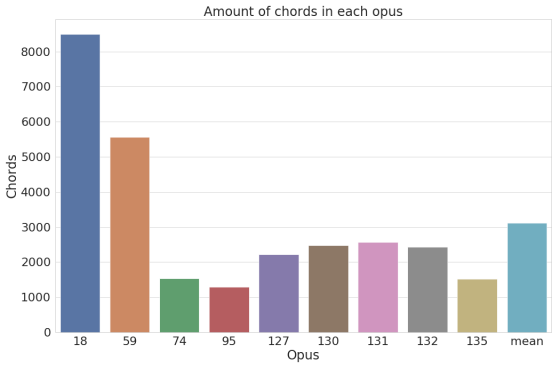

Figure 3. Amount of chords in each opus

3. RESULTS

Hyperparameter tuning for optimal n-gram length resulted in an optimal value of $n = 2$, which is consistent with findings in other modelling tasks in music reporting values between 2 and 4 (see e.g. [9, 16]). Thus, the simplest model actually achieved the best average accuracy score of 0.1952 (SD=0.024). While the average accuracy for $n = 3$ and $n = 4$ did not decrease substantially, results for larger n drastically decreased. The best recorded performance was 0.2372 for op. 130, and the lowest score of 0.1594 was achieved for op. 135. The accuracies for all opera are shown in Figure 2.

As for the LSTM model, it was observed that longer sequence lengths l in training only increased computational time at no substantial increase in accuracy, leading us to use the minimal value tested ($l = 10$) for prediction. An average accuracy of 0.1958 (SD=0.026) was obtained with a maximum of 0.2257 on op. 74 and a minimum of 0.1646 on op. 132. (see Figure 2). The accuracy values for both methods and all opera are reported in Table 4. The correlation between the two model accuracies is 0.21 and thus relatively weak.

4. DISCUSSION

The n-gram and LSTM models have similar mean accuracies and standard deviations. The weak correlation between the n-gram and LSTM model suggests that the accuracy of the models is indicative for certain properties of the data. Finding out what these properties are is not only an interesting musicological research question but will also allow to improve computational models for harmony prediction in the future. Specifically, opp. 135, 95, 131, and 127 have the lowest accuracy values, which suggests that harmonic progressions within these opera are especially hard to predict.

Having a more detailed look at the performance for each opus highlights the differences (see Figure 2). For instance, for op. 95 the LSTM model demonstrates substantially better performance than the n-gram model. Op. 95 is known as one of Beethoven's most experimental works about which he stated that "this work is written for a small circle of connoisseurs and is never to be performed in public" [22]. On the other hand, in opp. 18, 74, and 135 the n-gram model outperforms the LSTM model. A better understanding of where these differences originate from is an important step and will be pursued in future research.

The best performing n-gram model was of length $n = 2$, which contrasts with the musicological insight that harmonic dependencies can be highly non-local. This suggests that n-gram models, which are constructed to use local context information as much as possible (even those using more advanced smoothing and backoff methods) are not able to capture long-term dependencies in harmonic progression.

LSTM models, on the other hand, are supposed to capture long-term dependencies. The fact that, overall, the LSTM model does not outperform the n-gram model on

the present data set suggests that this potential was not fully leveraged as yet. One possible reason for this might be the representation of harmonies as simple string tokens, which does not make the rich structure of the harmonic annotations in the corpus accessible to the model.

Overall, an accuracy score of 19.5% is comparably low, which is most probably due to the rich annotation format in full Roman numeral representation making the annotated Beethoven corpus a particularly challenging data set to model.

5. CONCLUSION

We have evaluated two of the most commonly used models for sequence prediction, n-gram models and LSTM, on a recent published data set with harmonic annotations of Beethoven string quartets (ABC). Our LSTM model and the best performing n-gram model (with $n = 2$) showed comparable performance with an average accuracy of 19.5% over an alphabet of 800 harmonic symbols. The context length of $n = 2$ suggests that neither of the models was able to pick up on non-local dependencies in harmonic progressions, which underlines the importance of incorporating structural knowledge from music theory into computational models.

As the ABC dataset is largely unexplored and is unique due to its rich annotation format, we hope that our results – especially the accuracy score of 19.5% – provide a useful baseline for other researchers in the community.

Acknowledgments

MR would like to thank Mr Claude Latour for supporting this research.

6. REFERENCES

[1] D. B. Huron, Sweet Anticipation: Music and the Psychology of Expectation. MIT press, 2006.

[2] M. T. Pearce and G. A. Wiggins, "Auditory expectation: The information dynamics of music perception and cognition," Topics in cognitive science, vol. 4, no. 4, pp. 625–652, 2012.

[3] M. A. Rohrmeier and S. Koelsch, "Predictive information processing in music cognition. A critical review," International Journal of Psychophysiology, vol. 83, no. 2, pp. 164–175, 2012.

[4] M. Pearce and M. Rohrmeier, "Music cognition and the cognitive sciences," Topics in cognitive science, vol. 4, no. 4, pp. 468–484, 2012.

[5] L. B. Meyer, Emotion and Meaning in Music. University of Chicago Press, 2008.

[6] M. M. Farbood, "A parametric, temporal model of musical tension," Music Perception: An Interdisciplinary Journal, vol. 29, no. 4, pp. 387–428, 2012.

[7] M. Rohrmeier and P. Rebuschat, "Implicit learning and acquisition of music," Topics in cognitive science, vol. 4, no. 4, pp. 525–553, 2012.

[8] J. R. Saffran, E. K. Johnson, R. N. Aslin, and E. L. Newport, "Statistical learning of tone sequences by human infants and adults," Cognition, vol. 70, no. 1, pp. 27–52, 1999.

[9] M. Rohrmeier and T. Graepel, "Comparing feature-based models of harmony," in Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval. Citeseer, 2012, pp. 357–370.

[10] F. Colombo, S. P. Muscinelli, A. Seeholzer, J. Brea, and W. Gerstner, "Algorithmic composition of melodies with deep recurrent neural networks," arXiv preprint arXiv:1606.07251, 2016.

[11] F. Colombo, A. Seeholzer, and W. Gerstner, "Deep artificial composer: A creative neural network model for automated melody generation," in International Conference on Evolutionary and Biologically Inspired Music and Art. Springer, 2017, pp. 81–96.

[12] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier, "The annotated beethoven corpus (ABC): A dataset of harmonic analyses of all beethoven string quartets," Frontiers in Digital Humanities, vol. 5, jul 2018. [Online]. Available: https://doi.org/10.3389/fdigh.2018.00016

[13] A. Graves, "Generating sequences with recurrent neural networks," arXiv preprint arXiv:1308.0850, 2013.

[14] S. T. Piantadosi, "Zipf's word frequency law in natural language: a critical review and future directions." Psychonomic bulletin & review, vol. 21, no. 5, pp. 1112–30, oct 2014.

[15] D. H. Zanette, "Zipf's law and the creation of musical context," Musicae Scientiae, vol. 10, no. 1, pp. 3–18, 2006.

[16] M. T. Pearce and G. A. Wiggins, "Improved methods for statistical modelling of monophonic music," Journal of New Music Research, vol. 33, no. 4, pp. 367–385, 2004.

[17] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber et al., "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[18] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using blstm networks," arXiv preprint arXiv:1712.01011, 2017.

[19] S. Skuli, "How to generate music using a lstm neural network in keras," no. Dec 7, 2017. [Online]. Available: https://bit.ly/2IZtgm0

[20] F. Chollet, J. Allaire et al., "Keras," https://github.com/keras-team/keras, 2019.

[21] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv:1609.04747, 2016.

[22] B. Cooper, Beethoven. Oxford University press, 2000.