Evaluating Algorithms that Learn how to Compose Music from Scratch

New long– and short–term metrics for evaluating model-generated compositions

James Owers

A thesis presented for the degree of
Doctor of Philosophy

Supervised by:
Amos Storkey
Mark Steedman

University of Edinburgh, UK

January 2021

I, James Owers, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Evaluating whether creative content generated by a computer is 'good,' be it music, images, or text, is unsolved and not even well defined. We identify a property of music which is not modelled well, and propose new evaluation metrics for music generation which can be used to distinguish between real and generated data, and thus be useful for automatic quantitative analysis of generation quality.

We focus on symbolic music because ...TODO... This is interesting because ...TODO... and it has implications for ...TODO...

Finally, we make recommendations for how to make progress with respect to music generation and related tasks.

Acknowledgements

Interdum et malesuada fames ac ante ipsum primis in faucibus. Aliquam congue fermentum ante, semper porta nisl consectetur ut. Duis ornare sit amet dui ac faucibus. Phasellus ullamcorper leo vitae arcu ultricies cursus. Duis tristique lacus eget metus bibendum, at dapibus ante malesuada. In dictum nulla nec porta varius. Fusce et elit eget sapien fringilla maximus in sit amet dui.

Mauris eget blandit nisi, faucibus imperdiet odio. Suspendisse blandit dolor sed tellus venenatis, venenatis fringilla turpis pretium. Donec pharetra arcu vitae euismod tincidunt. Morbi ut turpis volutpat, ultrices felis non, finibus justo. Proin convallis accumsan sem ac vulputate. Sed rhoncus ipsum eu urna placerat, sed rhoncus erat facilisis. Praesent vitae vestibulum dui. Proin interdum tellus ac velit varius, sed finibus turpis placerat.

Table of Contents

List of Figures

List of Tables

Abbreviations

☐ check first instances of all these are spelt out, then all subsequent are not

API	**A**pplication **P**rogramming **I**nterface
AMT	**A**utomatic **M**usic **T**ranscription
JSON	**J**avaScript **O**bject **N**otation
MDTK	**M**idi **D**egradation **T**ool**k**it
NLP	**N**atural **L**anguage **P**rocessing
SOTA	**S**tate **o**f **t**he **A**rt

Chapter 1

Introduction

1.1 Target problems addressed in this thesis

1.2 Historical background

...TODO... Give historical background

1.2.1 First instance of music generation

...TODO... From Section 1.2 (Briot et al. 2019)

> The first music generated by computer appeared in 1957. It was a 17 seconds long melody named "The Silver Scale" by its author Newman Guttman and was generated by a software for sound synthesis named Music I, developed by Mathews at Bell Laboratories

1.2.2 Mozart using mechanical aids for idea generation

...TODO... From footnote 7 in Section 1.2 (Briot et al. 2019)

> One of the first documented case of stochastic music, long before computers, is the Musikalisches Wurfelspiel (Dice Music) by Wolfgang Amadeus Mozart. It was designed for using dice to generate music by concatenating randomly selected predefined music segments composed in a given style (Austrian waltz in a given key).

1.2.3 Ada lovelace noting computers could generate music

...TODO... From (Hollings et al. 2018) Ada Lovelace, "Sketch of the Analytical Engine invented by Charles Babbage, Esq., by L. F. Menabrea," Scientific Memoirs, vol. 3, ed. Richard Taylor, 1843, pp. 666-731 (this quote on p 694).

> "Note G" is the culmination of Lovelace's paper, following many pages of detailed explanation of the operation of the Engine and the cards, and of the notation of the tables. The paper shows Lovelace's obsessive attention to mathematical details - it also shows her imagination in thinking about the bigger picture.

> Lovelace overseed a fundamental principle of the machine, that

> the operations, defined by the cards, are separate from the data and the results. She observed that the machine might act upon things other than numbers, if those things satisfied mathematical rules.
>
> Supposing that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.

Lovelace also has the Lovelace Test of Creativity attributed to her - see (Ariza 2009).

1.3 Modern interest and achievements

...TODO...

- Imogen Heap: How AI is helping to push music creativity
- The AI Song Contest - In the AI Song Contest teams of musicians, artists, scientists and developers take on the challenge of creating a new Eurovision-like hit with the help of artificial intelligence.
- Swooshes, Seaboards, Synths and Spawn
- David Rosen and Scott Miles on the Neuroscience of Music and Creativity
- AI Music Generation Challenge 2020 - (Sturm 2020)

1.4 What are algorithms that learn

...TODO... define/introduce machine learning

1.5 What is composing music

...TODO...

1.6 What does it mean to compose from scratch

...TODO... what is the minimum information we supply as a starting point? What feedback do we give?

1.7 Motivation for this work

...TODO...

- why are we focussing on metrics and not human evaluation
  - how do we benchmark without them?
- where is the gap - not many metrics are available
- why do we care about 'from scratch'

(Sturm 2017) gives background as to why we need metrics but no specific methods.

Why do we need computational rather than human analysis

(Marsden 2016)

Computational music analysis needs to carve out a place for itself where it is not simply mimicry of human analysis, but a place which is not so distant from the human activity to prevent useful communication with musicians. We need to recall the potential value of computational analysis, the reasons we embark on this enterprise at all.

Metrics - more audio features than symbolic.

(Giraud et al. 2016)

There is less work to date that focuses on segmentation of symbolic scores

1.8 Scope of this work

...TODO... Symbolic music only

1.9 List of contributions in this thesis

...TODO...

- ☐ We provide a literature review of the current state-of-the-art with respect to algorithmic music composition: the challenges addressed and models presented

- ☐ Note that (Briot et al. 2019) does not include `COCOnet` nor any Transformer models so details of these is a new contribution

☐ Consistent open source python re-implementations of all models compared

Chapter 2

Literature Review

...TODO...

- ☐ Add content and TODOs from Quip:
  - ☐ https://quip.com/vVjVADMamfDm/2-Literature-Review
  - ☐ https://quip.com/v0MOAQGvMH3O/Literature-Review-Org-Notes
  - ☐ move data and notes tables in ./tables (use YAML, allows large text blocks for notes easy to convert to table with python)
- ☐ outline the different problems people currently try to / can solve, and how these problems relate to 'being able to compose'
- ☐ Review available metrics
  - ☐ Motivate the need for automated evaluation metrics
  - ☐ Why has there been more work on audio than symbolic?
  - ☐ Motivate the need for better evaluation for both short and long term by highlighting shortcomings for each method reviewed

- ☐ Describe state-of-the-art generative models for music composition
- ☐ Identify a gap with respect to modelling long term dependencies by outlining claims and proof of them thus far - this is a specific thing we are going show is poorly evaluated
- ☐ Inform the reader about the multitude of different ways we can represent music and their relative strengths and weaknesses
- ☐ Address ethical shortcomings with respect to learning to compose
- ☐ Update bibtex references to non-arxiv reference if available

2.1 Challenges addressed in the literature and how they are evaluated

2.2 A summary of evaluation methods for creative models

...TODO... how to evaluate generative models with a focus on music - how do people evaluate their success

2.2.1 The need for automated metrics

- ☐ Humans are expensive - show some efforts
- ☐ Humans do not agree - show some research proving this
- ☐ Humans are susceptible to change their opinion depending on context - show some research proving this

- ☐ Consistency is key when tracking performance over the long term
- ☐ Attempts to unify metrics and human opinion - give WMT as an example (Haddow 2020)

2.2.2 Differences between evaluating audio and symbolic outputs

(Dhariwal et al. 2020)

2.2.3 The impossible task of satisfying all evaluation requirements with a single metric

- ☐ Should tie a metric with performance for an intended task (Theis et al. 2016)

2.2.4 Evaluation metrics and representations used for extracting musical structures

2.3 Models for composing music

...TODO... State caveats about our distinctions:

1. Learning is essentially copying
2. By specifying the method of learning, we are incorporating expert knowledge

3. All models must work with a human composer to some extent - the programmer must choose a representation for the music and is therefore a composer in some senses

- ☐ Make and curate comparison table of models:
  - ☐ keep as csv
  - ☐ at min, we can use pandas to read and auto convert for insertion here
  - ☐ is there a way to `@reference` the file here and have pandoc insert? <- **do not spend time on this, cursory google!**
- ☐ Music Transformer (Huang et al. 2019)
- ☐ MuseNet (Payne 2019)
- ☐ Extend table from Chapter 7 and information from Chapter 6 in review paper (Briot et al. 2019)
- ☐ Go through https://paperswithcode.com/task/music-generation

2.3.1 Models which learn from scratch

...TODO... These are the models which our research pertains to

2.3.2 Models which do not learn from scratch

...TODO... These models are stated to highlight why they are different, have an unfair advantage in certain contexts, or explain why they are out of scope with respect to the investigation of this thesis.

2.3.2.1 Heuristic models which primarily copy and edit music from a database

2.3.2.2 Models which incorporate expert knowledge into their design

…TODO… e.g. with respect to structural hierarchy

2.3.2.3 Models which only work in conjunction with a human composer

2.3.2.4 Proprietary models

Models for which adequate details of their design are not publicly available

2.4 Methods for representing music on a computer

…TODO… how to represent music data - (in relation to 'from scratch,' what is the minimal information supplied to the models, and is there evidence of what difference it makes (either by experiment or just by reasoning?)

- ☐ Note our desires with respect to our modelling challenges: we want the input to be *minimal and flexible* - the model should learn as much as possible as if it were a human listener

- ☐ Ideally we would work directly on sound, but this involves an additional layer of representation.

2.4.1 Information that must be captured about a musical performance

2.4.2 The different representations of symbolic musical information

2.4.2.1 Summary of differences

- ☐ Highlight where *information* captured by each representation is both **different** and **more/less amenable to being learned**

2.4.3 Availability of data for each representation

- ☐ Quantity,
- ☐ Quality,
- ☐ Legal issues

2.4.4 Availability of software for different representations

- ☐ Describe MusPy (Dong et al. 2020) for conversion between data formats

- ☐ Describe Music21 (Cuthbert & Ariza 2010) for conversion between data formats

2.4.5 Evidence from the literature regarding modelling performance differences

- ☐ Find any reviews (or lack thereof) of model performance differences with respect to:
  - ☐ evaluation metrics
  - ☐ speed

2.5 Ethical considerations when designing automated methods for composing music

Chapter 3

New metrics for Evaluating Musical Generations

In this chapter we discuss features and metrics currently used to evaluate models which generate music, and the novel additions introduced in this thesis.

3.1 Features of symbolic music

- ☐ Discuss different features which can be extracted

3.2 Evaluation via downstream task performance

- ☐ Describe tasks as proposed in (McLeod et al. 2020) which could be used to evaluate models which compose

3.3 A phrase-level metric for short-term structure

3.4 A piece-level metric for long-term structure

Chapter 4

Data augmentation

> …I have always been of opinion that consistency is the last refuge of the unimaginative: but have we not all seen, and most of us admired, a picture from [Whistler's] hand of exquisite English girls strolling by an opal sea in the fantastic dresses of Japan?

–Oscar Wilde, "The Relation of Dress to Art: A Note in Black and White on Mr. Whistler's Lecture," Pall Mall Gazette, February 28, 1885.

A model's generalisability is its most important quality with respect to predicting its future performance in a new data context; we can expect a model which generalises well to perform *according to its evaluation* within a context nominally similar to the data from which its parameters were learned – the training dataset. A model which does not generalise well could still report very high evaluation metrics on the training dataset but perform poorly when released into the real world; its parameters having been overfitted to the training dataset.

Many strategies are employed to encourage models to generalise. For instance, one strategy, regularisation, controls the complexity of the model which can be learned. The idea is that simple models can generalise better; if a model is too powerful or complex it could simply learn the training dataset "by rote" and, since it has learned no abstract patterns, do nothing but regurgitate parts of the training dataset. A model can be regularised by adding a term to its loss function to penalise extreme settings of parameters i.e. changing the likelihood of learning a given set of parameters.

Data augmentation encompasses methods which aim to improve generalisability by effectively expanding the size of, and/or adding noise to, the training dataset seen by the model. By seeing a larger quantity of more variant data than the training dataset the model will better generalise – if there is randomised noise, the model cannot learn "by rote." This approach has had much success for NLP, in particular see these recent approaches: BART (Lewis et al. 2020) and, subsequently, Speller100 (Lu et al. 2021).

In this chapter we present a toolkit for adding noise to symbolic music: the MIDI degradation toolkit (MDTK). This is previously work published work, jointly authoured by the author of this thesis and a collaborator, presented at the 21st International Society for Music Information Retrieval Conference.

4.1 The MIDI degradation toolkit

In (McLeod et al. 2020) we presented MDTK. It was originally developed for an Automatic Music Transcription (AMT) setting, specifically, to generate large datasets for training discriminative models which correct transcription errors. MDTK is a python (Rossum 1995) package which contains, amongst other things, functions which alter the symbolic music data input.

The paper, poster, and a short introductory video are available online here: https://program.ismir2020.net/poster_6-10.html

The toolkit code is open source, and available on GitHub here: https://github.com/JamesOwers/midi_degradation_toolkit

4.1.1 The degradations available in MDTK

Figure 4.1

We see in 4.2.

Figure 4.3 shows.

4.2 An experiment illustrating performance gains with MDTK

Figure 4.1: Illustrations for all the degradations currently available in MDTK

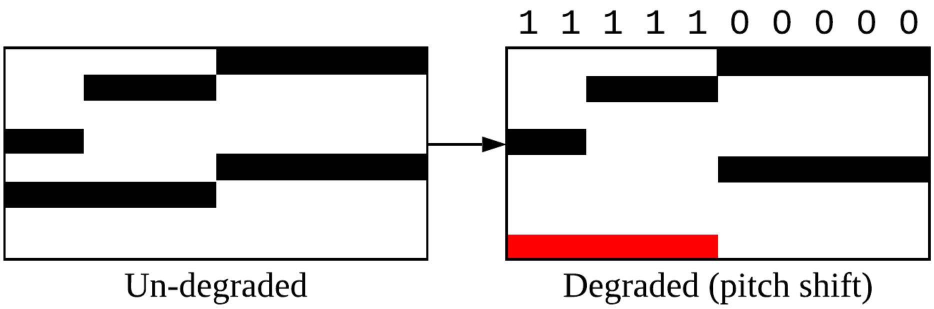

Figure 4.2: An example degradation performed by MDTK

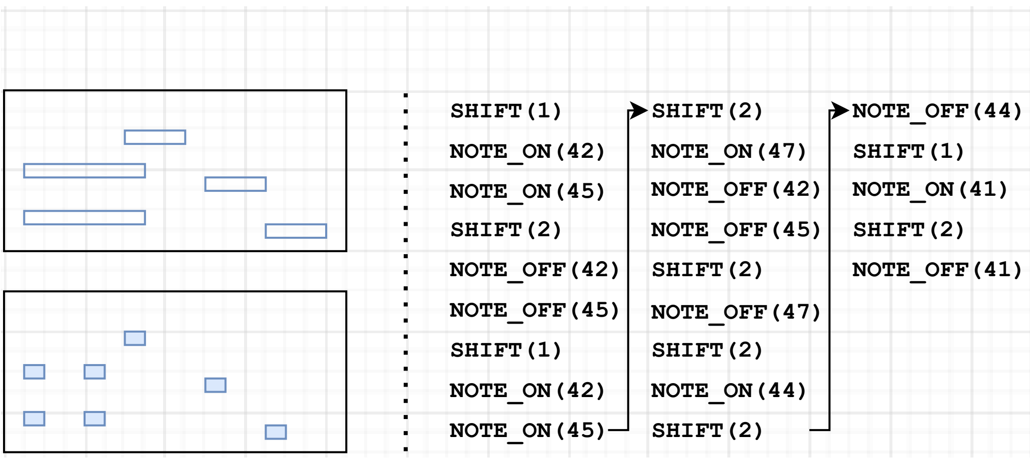

Figure 4.3: Data formats.

Chapter 5

Evaluating State-of-the-Art Music-generating Models

- ☐ If the main contribution is evaluating the long term structure, then ensure this is emphasized either in the title of this chapter or in the first lines

5.1 Comparative analysis using new and existing metrics

- Use phrase and piece level metrics to evaluate state-of-the-art models
- Compare and contrast, outlining the issues identified (e.g. meandering, no high-level structure)

5.2 Strengths and shortcomings of existing models

5.3 Avenues for improvement

Chapter 6

Conclusion

Chapter 7

A New Model

Potential ideas:

- An improved generative model for music
  - Training like BERT? http://jalammar.github.io/illustrated-bert/
  - Using mdtk for data augmentation in training (negative examples?), making them more robust
  - Alternative training objectives:
    * crossentropy slow and not musically informed
    * can we use something akin to word error rate (this has been done for text)
- Alternative ways to encode music: encoding chords and phrases in a low-rank continuous space
  - Have done some work on this with convnets and generating continuations
    * low rank was enforced by cross-product ing two vecs

– Could investigate effect of different representations for music on performance

Chapter 8

References

…TODO…

- ☐ check over using https://www.cl.cam.ac.uk/~ga384/bibfix.html
- ☐ also check with https://github.com/yuchenlin/rebiber
- ☐ Check all title casing correct (use curly braces around letters which should remain as they are). All titles should be in Title Case.

Ariza, C., 2009. The Interrogator as Critic: The Turing Test and the Evaluation of Generative Music Systems. *Computer Music Journal*, 33(2), pp.48–70. Available at: https://www.jstor.org/stable/40301027 [Accessed January 22, 2021].

Briot, J.-P., Hadjeres, G. & Pachet, F.-D., 2019. Deep Learning Techniques for Music Generation – A Survey. *arXiv:1709.01620 [cs]*. Available at: http://arxiv.org/abs/1709.01620 [Accessed January 22, 2021].

Cuthbert, M.S. & Ariza, C., 2010. music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*. Available at: https://dspace.mit.edu/handle/1721.1/84963 [Accessed January 23, 2021].

Dhariwal, P. et al., 2020. Jukebox. *OpenAI*. Available at: https://openai.com/blog/jukebox/ [Accessed February 19, 2021].

Dong, H.-W. et al., 2020. MusPy: A Toolkit for Symbolic Music Generation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. Montreal, Canada. Available at: https://program.ismir2020.net/static/final_papers/187.pdf [Accessed January 26, 2021].

Giraud, M., Groult, R. & Levé, F., 2016. Computational Analysis of Musical Form. In D. Meredith, ed. *Computational Music Analysis*. Cham: Springer International Publishing, pp. 113–136. Available at: https://doi.org/10.1007/978-3-319-25931-4_5 [Accessed February 21, 2021].

Haddow, B., 2020. EMNLP 2020 Fifth Conference on Machine Translation (WMT20). *2020 Fifth Conference on Machine Translation (WMT20)*. Available at: http://www.statmt.org/wmt20/ [Accessed January 23, 2021].

Hollings, C., Martin, U. & Rice, A.C., 2018. *Ada Lovelace: The Making of a Computer Scientist*, Oxford: Bodleian Library.

Huang, C.-Z.A. et al., 2019. Music transformer: Generating music with long-term structure. In *7th international conference on learning representations, ICLR 2019, new orleans, LA, USA, may 6-9, 2019*. OpenReview.net. Available at: https://openreview.net/forum?id=rJe4ShAcF7.

Lewis, M. et al., 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. Online: Association for Computational Linguistics, pp. 7871–7880. Available at: https://www.aclweb.org/anthology/2020.acl-main.703.

Lu, J., Long, J. & Majumder, R., 2021. Speller100 expands spelling correction technology to 100+ languages. *Microsoft Research*. Available at: https://www.microsoft.com/en-us/research/blog/speller100-zero-shot-spelling-correction-at-scale-for-100-plus-languages/ [Accessed March 7, 2021].

Marsden, A., 2016. Music Analysis by Computer: Ontology and Epistemology. In D. Meredith, ed. *Computational Music Analysis*. Cham: Springer International Publishing, pp. 3–28. Available at: https://doi.org/10.1007/978-3-319-25931-4_1 [Accessed February 17, 2021].

McLeod, A., Owers, J. & Yoshii, K., 2020. The MIDI Degradation Toolkit: Symbolic Music Augmentation and Correction. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. Montreal, Canada. Available at: https://program.ismir2020.net/static/final_papers/182.pdf [Accessed January 26, 2021].

Payne, C., 2019. MuseNet. *OpenAI*. Available at: https://openai.com/blog/musenet/ [Accessed January 23, 2021].

Rossum, G. van, 1995. Python tutorial. Available at: https://ir.cwi.nl/pub/5007 [Accessed February 21, 2021].

Sturm, B., 2020. AI Music Generation Challenge 2020. *The 2020 Joint Conference on AI Music Creativity*. Available at: https://boblsturm.github.io/aimusic2020/ [Accessed January 22, 2021].

Sturm, B.L.T., 2017. Benchmarking "music generation systems?" *Folk the Algorithms*. Available at: https://highnoongmt.wordpress.com/2017/03/19/benchmarking-music-generation-systems/ [Accessed February 18, 2021].

Theis, L., Oord, A. van den & Bethge, M., 2016. A note on the evaluation of generative models. In Y. Bengio & Y. LeCun, eds. *4th international conference on learning representations, ICLR 2016, san juan, puerto rico, may 2-4, 2016, conference track proceedings*. Available at: http://arxiv.org/abs/1511.01844.