Interacting with GPT-2 to Generate Controlled and Believable Musical Sequences in ABC Notation

Anonymous ACL submission

Abstract

Generating symbolic music with language models is a promising research area, with potential applications in automated music composition. Recent work shows that Transformer architectures can learn to generate compelling four-instrument scores from large MIDI datasets. In this paper, we re-train the small (117M) GPT-2 model with a large dataset in ABC notation, and generate samples of single-instrument folk music. Our BLEU and ROUGE based quantitative, and survey based qualitative, evaluations suggest that ABC notation is learned with syntactical and semantic correctness, and that samples contain robust and believable n-grams.

1 Introduction

Recent advances in deep learning have greatly improved the performance of neural generative systems at automatic music generation. For example, Magenta's MusicVAE (Roberts et al., 2018) uses hierarchical autoencoders to interpolate novel music samples between different points in a MIDI latent representation. Similar techniques have been proposed for the task of learning language models, mostly in Natural Language Processing (NLP). For example, the Transformer-based neural architectures of BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019), and Transformer XL (Dai et al., 2019) use encoders/decoders and various attention mechanisms to achieve great performance at language learning and generation. Therefore, it is no surprise that these models have been applied for learning and generating symbolic music scores, assuming that similar sequence-to-sequence attention mechanisms to those of written natural language hold for written music. For example, LakhNES (Donahue et al., 2019) and MuseNet (Payne, 2019) use these language models over MIDI music representations, successfully addressing large scale, multi-instrument, and long sequence MIDI score learning and generation.

```
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB |1 dBA AFD :|2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB |1 dBA ABd :|2 dBA AFD |]
```

Listing 1: An example tune in ABC notation.

However, a shortcoming of these works is that they learn exclusively over MIDI representations, leaving unanswered questions for other genera and datasets. For example, folk and traditional music are typically encoded using ABC notation (Walshaw, 2011). Moreover, such experiments are almost exclusively evaluated using perplexity (Brown et al., 1992) instead of other language evaluation metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). In this paper, we propose to address these issues by adapting the pre-trained small (117M parameters) language model of GPT-2 (Radford et al., 2019) to learn representations of an ABC notation dataset. ABC notation is an ASCII based character set code that facilitates the sharing of music online (see Listing 1). The first lines indicate the tune index in the file (X:); title (T:); time signature (M:); default note length (L:); type of tune (R:); and key (K:). Following this is the tune, with the | symbol separating measures. Notes are displayed with the letters a to g, where lowercase letters and apostrophes denote higher octaves and uppercase letters and commas denote lower octaves. Further punctuation marks represent variations in the tune. We use conditional sampling, feeding the model two measures and letting it generate the sequence remainder. We evaluate these samples quantitatively, using the BLEU and ROUGE metrics in various n-gram tests for robustness; and qualitative, via a user survey. Our research question is: "To what extent can language models learn robust representations of ABC notation single-instrument folk music?".

2 Related Work

Many language models derived from results in computer vision have been investigated in recent years, most with successful applications in music learning and generation. For example, long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) recurrent models are commonly used for text generating tasks; and hidden Markov models (HMM) (Rabiner and Juang, 1986) have been used for e.g. speech recognition. More recently, advances in encoder/decoder neural architectures have produced so-called Transformer models, like BERT (Devlin et al., 2018); OpenAI's GPT-2 (Radford et al., 2019) –a sequence to sequence transformer with an attention mechanism; and Transformer XL (Dai et al., 2019), a high performance transformer with high compute requirements. The application of these models to music generation has produced various results. For example, OpenAI's Jukebox (Dhariwal et al., 2020) procudes high-fidelity music in the raw audio domain. However, we consider here the language models that can be applied to *symbolic* music generation. In this area, MusicVAE (Roberts et al., 2018) uses a hierarchical variational autoencoder to learn an interpolable latent space of MIDI representations. The works closest to ours are MuseNet (Payne, 2019) and LakhNES (Donahue et al., 2019); in these, authors re-train a Transformer model pre-trained on the Lakh MIDI dataset (Raffel, 2016), a large collection of 176,581 unique MIDI files, to generate four-instrument scores. Our approach is inspired by these works, but focuses on: (a) using GPT-2 instead of Transformer XL, due to the former's excellent text generation capabilities and left-to-right training; and (b) learning ABC representations of folk and traditional music, rather than using cross-domain MIDI files.

3 Methodology

The process for this research began with cleaning the original data set[1] and putting all samples in separate files. This data set was then used to fine-tune the GPT-2 model on. The fine-tuning is stopped, when the loss barely decreases over a large amount of time. This final model will be used to create conditional samples by feeding the model a short musical sequence of two measures from an existing song and letting it generate a subsequent sequence. From the output, another two measures are taken. The two measures from the original song and the generated part are combined to form the new input sequence. This process is repeated, alternating measures from the original song with measures that are generated by GPT-2. Then, these samples are evaluated on their syntax and semantics and they are evaluated using BLEU, ROUGE and a user evaluation form. The outcome of these evaluations will determine whether valid, but also fluent musical pieces can be generated, by having some control over the process.

4 Experiment

The model with 117M parameters was used for this, considering the limited amount of time and the fact that larger models might overfit. Furthermore, the longer the model is trained, the better it can familiarize itself with the training data. This does not necessarily mean it performs better when generating output, but it does increase the chances, up to a certain point. This is why the training is stopped when the loss hardly decreases over a substantial amount of time. The model alternated between an average cross-entropy loss of 0.86 and 0.94 over several hours, meaning the model had a hard time optimizing further from this point on. The resulting model was used to generate controlled sequences of music. Two songs from the used data set were chosen and two songs from the left out data set were chosen to diversify. Firstly, the first two measures of an original song are fed, including the header. Based on this, the model is then prompted to generate notes that follow the sequence. From the outcome, only the first two measures are added to the input. The resulting, larger sequence will be fed to the model again, so it can extend this sequence with two measures as well. This is repeated three times, to obtain a song of 12 measures, that consists of 6 measures from the original song and 6 measures generated by the model, alternately.

[1] See https://www.gwern.net/GPT-2-music

4.1 Quantitative Evaluation

The similarity between the original melodies and the samples are calculated using the BLEU and ROUGE metrics. Two tables are displayed for the n-grams of BLEU and ROUGE scores for each sample.

BLEU scores				
	1-gram	2-gram	3-gram	4-gram
Sample 1	0.60	0.51	0.48	0.46
Sample 2	0.71	0.57	0.48	0.45
Sample 3	0.56	0.47	0.44	0.42
Sample 4	0.76	0.60	0.54	0.52

Table 1: The BLEU scores for all samples over n-grams 1 to 4

ROUGE scores		
	1-gram	2-gram
Sample 1	0.62	0.53
Sample 2	0.72	0.58
Sample 3	0.89	0.74
Sample 4	0.77	0.60

Table 2: The ROUGE scores for all samples over n-grams 1 to 4

The BLEU score measures how many bi-grams from the GPT-2 generated samples occur in the orginal song. The scores can range from 0 to 1. 0 indicating no overlap with the original song, 1 indicating a perfect overlap with the original song. Since, half of a sample is copied from the original song, the precision should not go much below 0.50. However, this might occur, when the generated sample has less tokens than the original song, which is the case in sample 3. Samples 1 and 2 have some, but not excessive overlap with their originals. While the fourth sample has many overlapping bi-grams with the original song. The ROUGE score computes the number of bi-grams from the original song that occur in the generated sample. Samples 1, 2 and 4 overlap a little more than 50%, keeping in mind that this might be caused by the length of the sample. Sample 3 shows that numerous bi-grams overlap with the generated sample.

4.2 Qualitative Evaluation

The questionnaire yielded 83 responses. Roughly half of these were male and half were female, with one person preferring not to specify this. Slightly more than 50% of the participants were between the age of 10 and 25, while the rest was older. Most candidates were educated on the level of a Bachelor's degree. About a quarter is educated higher than this and the remaining quarter is educated lower or not at all. 52% of participants were students, of which 12% had either a full-time or part-time job as well. Another 41% was occupied by solely a full-time job, while the remaining percentage either had a part-time job, was unemployed or had another occupation. As expected over half of the participants were Dutch. The other nationalities are spread over 15 other countries. As for the musical knowledge, half of the participants scored themselves below average, approximately 20% thought they were (close to) an expert and over a quarter thought they had an average level of musical knowledge.

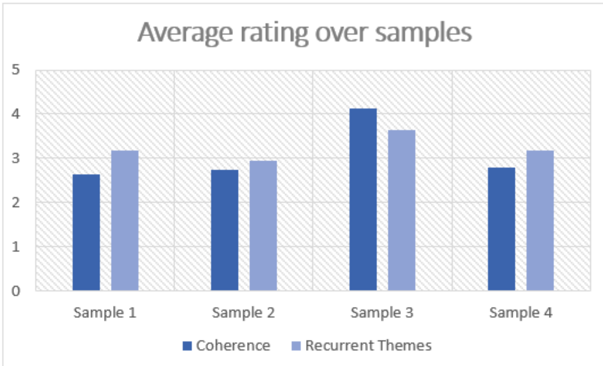

Figure 1: The average ratings of the questionnaire by sample

Regarding the scoring of the samples, the questions were answered by a rating from 1 to 5. The first sample got an average scoring of 2.6 for coherence and 3.1 for the amount of recurrence. The second sample got a coherence of 2.7 and was scored 2.9 for recurrence. The third sample had a coherence of 4.1 and a recurrence of 3.6. The fourth sample had a coherence of 2.6 along with a scoring of 2.5 for recurrent themes. The two samples that contained existing songs got a score of 3.7 and 3.9 for coherence and a score of 2.9 and 3.5 for recurrence. [2]

4.3 Syntax and semantics

The samples are presented, where the areas in bold are generated by GPT-2. When looking at the syntax of the first sample, the model seems to have

[2]See https://soundcloud.com/user-512999768

```
X:129531
M:6/8
K:Cmaj
^c2^A^G^G^G|^c^A^A^A2^G|

K:Cmaj
|:CDECDE|=F2GA2G|

|^c2^A^G^G^G|^c^A^A^A2^G|

M:3/4
K:Cmaj
|:=C=B,=C=F=G,=C|=B,=D=D=F2=G|

|=f2^g=f^c^d|=f^c^A^A2^G|

=A=G=E=c2|1=E=C=B,=D=C|
```

Listing 2: The first sample of GPT-2's generated ABC notation.

```
X:129557
M:12/8
K:Cmaj
|^C2=F^G2^G |^A^c^A^G=F^D|

=F=E/2=D/2=C=F=G=A|=G=F=D=F2=A,|

^C2=F^G2^G|^A^c^A^G=F^G|

L:1/8
K:Gmaj
|:D2G2GF|DEGABc|

=f^d^c=c^A^G|^A^c^A^G=F^G|

M:6/8
K:Cmaj
|:^C^D=F^C^G^F|^G^C^c^G^F^A|
```

Listing 3: The second sample of GPT-2's generated ABC notation.

adopted it well. However, when looking at the semantics, the meter is difficult for the model to adhere to. As for the meter in this melody, which is 6/8, the model mostly gets it right, until it changes the meter to 3/4. After this, the model still holds on to the first meter and in the last generated part follows neither of the two meters. Furthermore, it seems that the model wants to specify what key and meter it is using, even though the key is the same as the given key. What stands out is that the model is reluctant to use the caret, despite the fact that this symbol is frequent throughout the original song. On top of this, the model seems to have a tendency to use equality signs, which represents an unaltered pitch of a note. The melody of sample 2 is syntactically flawed. A colon is used to open a repetition, however it is never closed. This happens in both the second and third generated parts. The meter is 12/8 in the beginning and changed to 6/8 in the last generation. The key is changed in the last two generations, first to G major and then back to C major. Another noticeable concept is that in the first generation the notes are all naturalized, while this is uncommon in the original song. However, the carets that are frequent are not adopted until the last generation.

```
X: 136
K:C
M:2/4
L:1/8
"C"ce/2c/2 ge/2c/2|"C"ce/2c/2 "G7"B/2c/2d/2e/2|

=C=E/2=F/2=G=c2|=c2

"C"ce/2c/2 ge/2c/2|"C"ce/2c/2 "G7"B/2c/2d/2e/2|

|AcedcB|GED2Bc|

"Am"ce/2c/2 "Em"Be/2B/2|"F"Ac/2A/2 "C"GA/2G/2

|1=B,=C/2=B/2=A^G=E=C|=D=E=F=G=A=B|
```

Listing 4: The third sample of GPT-2's generated ABC notation.

```
X: 155
M:4/4
L:1/4
K:F
"F"F>G A F|"Bb"D F "C"C2|

K:A
Acc/e/c/eea2|geBcdec|

"F"F/2 F G/2 A F|"Bb"D F "Am"C2|

K:Gmaj
G>EEBAGA|BcdA/2G/2[B3G3]B|

"Fmaj7"A c c c|A>c c2|

DBB>A|B2G>B|
```

Listing 5: The fourth sample of GPT-2's generated ABC notation.

5 Conclusion

Influencing the generation process of samples led to reasonable results. The model does not deviate far from correct syntax and semantics. Besides this, plausible results are obtained using both the BLEU and ROUGE metrics. This can be deducted from the small decrease in performance while the n-grams increase. The user evaluation showed around average or higher ratings for each of the samples obtained from users with different backgrounds. These results are reason to believe that this method can result in robust musical sequences. However, improvements may lead to better results. A larger dataset may increase the models pattern recognition. A dataset could also contain ABC Notation in another genre to see if one genre is easier to learn than others. Besides this, GPT-2 has a number of parameters that can be altered when creating samples. On the contrary, one might choose to use another language model altogether, such as those mentioned in the related work section.

References

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

Chris Donahue, Huanru Henry Mao, Yiting Ethan Li, Garrison W Cottrell, and Julian McAuley. 2019. Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training. *arXiv preprint arXiv:1907.04868*.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Christine Payne. 2019. MuseNet. `https://openai.com/blog/musenet/`.

Lawrence Rabiner and B Juang. 1986. An introduction to hidden Markov models. *ieee assp magazine*, 3(1):4–16.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Colin Raffel. 2016. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Ph.D. thesis, Columbia University.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*.

Chris Walshaw. 2011. The ABC music standard 2.1. Technical report, abcnotation.com.