CHORD2MELODY - A TRANSFORMER FOR MUSIC GENERATION

A PREPRINT

Toshiyuki Sakamoto
Freelance
tanrei@nama.ne.jp

December 14, 2020

ABSTRACT

The generation of music and the generation of text have some basic similarities. First, it is a time-serialized data, going from start to finish. Second, it consists of discrete data, with the basic units being, in the case of music, sound note for the pitch and length of the sound, and in the case of text, word. On the other hand, the generation of music and the generation of text differ in some respects: in music, different instruments are played at the same time, and multiple sounds are vocalized at the same time to construct a chord. In this paper, we attempt to see if it is possible to serialize notes, the basic unit of music, as well as words in text, and use the GPT-2 text generation model to generate music. We trained these models on a dataset of over 100,000 bars of pop music. We indicate that it is possible to generate music using the GPT-2 text generation model at different number of tracks (5 and 17 tracks). We also asked 9 volunteers to rate the music generated as an evaluation and show the difference between it and existing studies. All codes and trained models and generated music samples are available at 'https://github.com/tanreinama/chord2melody' .

Keywords Chord2Melody · Mugic Ceneration · Transformer · GPT-2

1 Introduction

Automatically generating content that would be created by human creators is a major technical challenge for AI. In recent years, GPT-2[Radford , 2019] has made great advance in text generation. In text generation, a sentence is constructed of a series of words, and sentence meaning is constructed from a series of word meanings. In addition, the context of document is constructed from a series of sentence meanings. Therefore, text generation needs to deal with multi-layered concepts. Similarly, in music, a melody consisting of a series of notes has a meta-concept of chord progressions. The chord progression corresponds to the sentence meaning in the text and determines the mood of the melody in an intermediate time range. Therefore, the generation of music also needs to deal with the concept of multilayeredness. The existence of this multilayered concept makes it difficult for models that can only generate music in a short time range to generate practical content. Therefore, a model that can handle longer time ranges and multilayered concepts is required.

1.1 Music

Music is time-serialized data, just like text. We are not dealing here with songs that contain lyrics, but with music played only by instruments. The basic unit of music is a note, which corresponds to a word in the text. Music has several elements that differ from text. Namely, melodic, accompaniment and rhythm. All instruments are synchronized in rhythm by being played at the right time to the beat provided by the percussion. Melodic and accompaniment are divided into roles in the instrument type tracks. The roles are variable and can be swapped. And instruments other than percussion have a note pitch. The sounds played by instruments with note pitch, whether melodic or accompanying, are loosely constrained by chord progressions. The chord progressions determines the general mood of the melody. The notes in the melodic or accompaniment are based on the octaves of the notes in the chord, but other notes can be included as well. However, there are certain constraints on notes that are not part of the notes in the octaves of the chord, because there are too many notes that are not part of the octaves of the chord, the melody sound dissonant. The

freedomness allowed within those constraints (time constraints by rhythm and note pitch constraints by chords) makes composition a creative task.

1.2 GPT-2

In recent years, significant advances have been made in natural language processing, and they use the Transformer model. For example, GPT[Alec , 2018], BERT[Jacob , 2019], and XLNet[Li , 2019]. In particular, GPT-2[Radford , 2019] has achieved significant results in the task of text generation. In content generation, the same partial data may be generated as the training data; for instance CIFAR-10 has a 3.3% overlap between train and test images[Barz , 2019]. An evaluation of GPT-2 by Radford et al. suggests that the size of the training data is important by assessing the overlap between the training data and the generated data[Radford , 2019].

2 Related Work

2.1 MuseGAN

MeseGAN is a music-generating neural network created by Hao-Wen Dong et al[Hao-Wen , 2017]. MeseGAN makes use of an Generative Adversarial Network (GAN). MeseGAN uses piano rolls. Piano roll is an image with the time horizontally, the pitch of the notes vertically, and the volume placed on the channel. MeseGAN handles music as four-dimensional data by stacking the piano rolls for each track of instrument. It then uses CNN to generate the music. MeseGAN is able to generate short pieces of music nicely. However, it is limited to short time range (4 bars). Also, since it does not handle the music as time-serialized data, there is no integrity between the previously generated fragments and the next generated fragments as a continuous piece of music.

2.2 Image-GPT

Image-GPT is a neural network that generates images created by Chen et al[Chen , 2020]. In Image-GPT, the pixels in an image are clustered into discrete data. Then, by scanning the image, the pixels are encoded into time-serialized data. The pixels that encoded to time-serialized data are input to the Transformer same as the GPT-2. Then it outputs is a sequence of pixels. This means the task of inputting the top half of the image and generating the bottom half. Image-GPT has shown that the GPT-2 model allows for the generation of a variety of time-serialized data.

3 Implementation

The model we use is the same as the GPT-2. Therefore, the key point is how to encode the musical elements, the notes, into time-serialized data and, the procedure for generating the music after training.

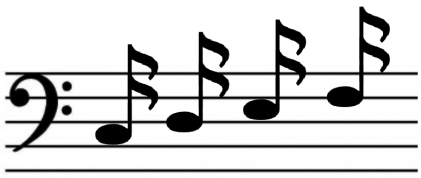
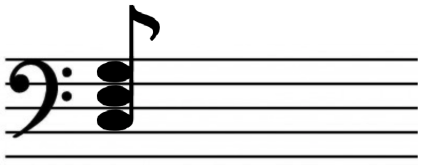
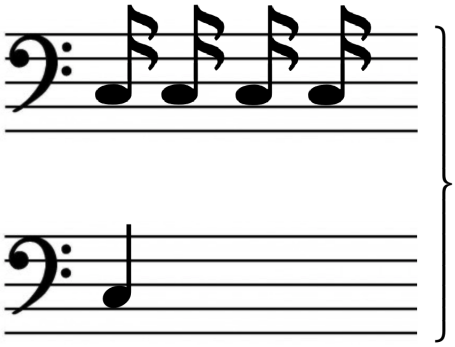
3.1 Serialization of notes

The pitch of notes in music is based on the fundamental frequency contained in the sound. In normal tuning, the center A (A4) will be 440 Hz. Then, an octave is divided into 12 notes to define the pitch of the sound. In the General MIDI, it defines the pitch of a note with 128 step values from C-1 (8.2Hz) to G9 (12543.9Hz). We only encode 84 notes from C1 (32.7Hz) to B7 (3951.1Hz), which are actually used in music.

Note encoding maps the combination of note pitch and the track to be played to individual tokens. The length of a note is represented by dividing the duration of the note by the time it is played and adding the required number of "time notes". The unit of time separation is a sixteenth note. For example, a sixteenth note of C2 height on track 0 is encoded as "12, <|timenote|>". A progression of sixteenth notes that C2, D2, E2, F2, on track 0 would be encoded as "12, <|timenote|>, 14, <|timenote|>, 16, <|timenote|>, 17, <|timenote|>". If track 0 contains the chords C2, E2, G2 in eighth notes, it would be encoded as "12, 16, 19, <|timenote|>, 12, 16, 19, <|timenote|>". This encoding scheme cannot distinguish between consecutive sixteenth notes of the same pitch and longer notes. That is, the four consecutive sixteenth notes (C2,C2,C2,C2) and the single quarter note (C2) on track 0 are both encoded as "12, <|timenote|>, 12, <|timenote|>, 12, <|timenote|>, 12, <|timenote|>" (Table1). Finally, there is an "<|endnote|>" which represents the end of the music.

After add "<|timenote|>" and "<|endnote|>", you get 422 types of tokens after encoding if you have 5 tracks, and 1430 types of tokens after encoding if you have 17 tracks. This is considerably less than the number of words (50256) used, such as in the GPT-2 text generation model. The tokens played at the same time are aligned so that the bass and piano tracks come first. This alignment is needed to generate the melody from the chord progression described below.

Table 1: Serialization Examples of notes

Notes	Serialized Data
	12, <|timenotel|>, 14, <|timenotel|>, 16, <|timenotel|>, 17, <|timenotel|>
	12, 16, 19, <|timenotel|>, 12, 16, 19, <|timenotel|>
	12, <|timenotel|>, 12, <|timenotel|>, 12, <|timenotel|>, 12, <|timenotel|>

3.2 Model Parameters

Our model is the same as the GPT-2-small. That is, 768 dimensions with 12 layers. Due to the small number of token types used (corresponding to the number of vocabularies in GPT-2), the actual number of parameters is considerably less than in GPT-2-small(117M): 86167296 for 5 track-model and 86941440 for 17 track-model. It is not optimized for music generation, as like GPT-2 is optimized for text generation. There is room for improvement; even with the same number of parameters, more layers and fewer dimensions are possible because of the fewer types of notes used.

3.3 Data Augmentation

In order to increase the total amount of data to be trained, Data Augmentation was introduced. Data Augmentation shifts the pitch of the notes in non-percussion tracks uniformly. In other words, it is a modulation.

3.4 Melody2Melody

Melody2Melody is a simple application of GPT-2 model in our implementation. The music given in a MIDI file is merged into 5 or 17 tracks, depending on the track of instrument, and encoded into a token sequence. The model then generates the continuation token sequence. The generated token sequence is again converted to a MIDI file and saved.

If there is no input, Merody2Merody can generate a completely new melody by continuation "<|endnotel|>". The model generates 1024 tokens at a time. To generate longer music, a quarter of the generated token sequence is used as input for the next generation task. This allows us to generate very long melodies with musical integrity.

3.5 Chord2Melody

Chord2Melody is an implementation for generating melodies from chord progressions, which was created by studying the process of human creators to compose music. The chord progressions are given externally as a parameter. A given chord progression is entered into the model as a chord to be played by the bass and piano tracks at the corresponding timing. The model handles the input chords as part of the music and generates their continuation. The generated melody is used until the next chord progression. In the next chord progression, the previously generated melody and the chords corresponding to the new chord progression are input to the model. The generation process is repeated as many as necessary. Chord2Melody can mix the timing to specify the chord progression and the timing not to specify it. If no chord progression is specified, the model generates the next token sequence as it is. You can also specify the chord progression only at the beginning of the music, and generate subsequent tokens. In this case, it is equivalent to inputting the output of Chord2Melody into Melody2Melody. In the absence of external input, Chord2Melody is identical to Melody2Melody.

4 Experiment and Results

4.1 Training

The Lakh Pianoroll Dataset (LPD)[1] was used for training; the LPD is a translation of the Lakh MIDI Dataset[Colin , 2016] into piano roll data. We used the LPD-FULL data, which contains 174154 music datas. In addition, Data Augmentation was performed to create a 6x data set. For training, we used Adam and ran 6.6M iterations with a learning rate of 5e-5. Training was performed on both LPD-5 with 5 tracks and LPD-17 with 17 tracks to make sure that the learning proceeded in the same way.

4.2 Generation

The GPT-2 model uses the top_k and top_p meta-parameters to select the generated token sequence. We found that the top_k and top_p meta-parameters have a significant effect on the generated melodies. When the value of the meta-parameter ia small, the generated melodies have a calm and peaceful tone. On the other hand, when the value of the meta-parameter is big, the generated melodies are dynamic and intense. Chord2Melody imposes constraints on the model in terms of when to specify the chord progression. If we increase the value of the meta-parameter when the constraint is given, the tune will be the same as when there is no constraint. We adopted "top_k=0, top_p=7" as the default meta-parameters, and "top_k=0, top_p=40" only for the timing of the restriction by chord progression.

4.3 Evaluation

For the evaluation of the generated music, 9 volunteers were asked to evaluate it directly. All comparisons were made with 5 tracks of data. For comparison, melodies extracted from LPDs and melodies generated by MuseGAN were added to the subject. To make the conditions fair, we extracted 4 bars of melody from the LPD from a random time location, not the beginning of the music. And Chord2Melody had no external input (generated as a continuation of "<lendnotel>"). Thus it is equivalent to generation by Melody2Melody. Melodies generated from Chord2Melody was extracted only 4 bars. Therefore, in all questionnaire, a melody of the same length is evaluated. In addition, the BPS of the melody was also changed to the same (120 bps). In the questionnaire, 3 musics were presented to the volunteers in each. The questionnaire consisted of 4 options: "good", "slightly good", "slightly bad", and "bad" music melodies, respectively. In order to eliminate preconceptions, the music numbers presented were rearranged in a random order.

4.4 Result

The results of the questionnaire are shown in Figure1, where lpd* is the melody extracted from LPD, mg* is the melody generated by MuseGAN, and c2m* is the melody generated by Chord2Melody. Naturally, LPD has the highest number of "good" and "slightly good" combined; number is 20, which is 74% of the total. In MuseGAN, that number is 13, which is 48% of the total. Chord2Melody has more variance by files than MuseGAN. There is one melody in particular (c2m3) that is extremely poorly rated; when c2m3 is excluded as an outlier, Chord2Melody is better than MuseGAN (12, which is 67% of the total). But when the average including c2m3 is taken, Chord2Melody is slightly worse than MuseGAN (12, which is 44% of the total). The mean and variance of each alternative divided by technique are shown in the Figure2.

[1] https://salu133445.github.io/lakh-pianoroll-dataset/dataset.html

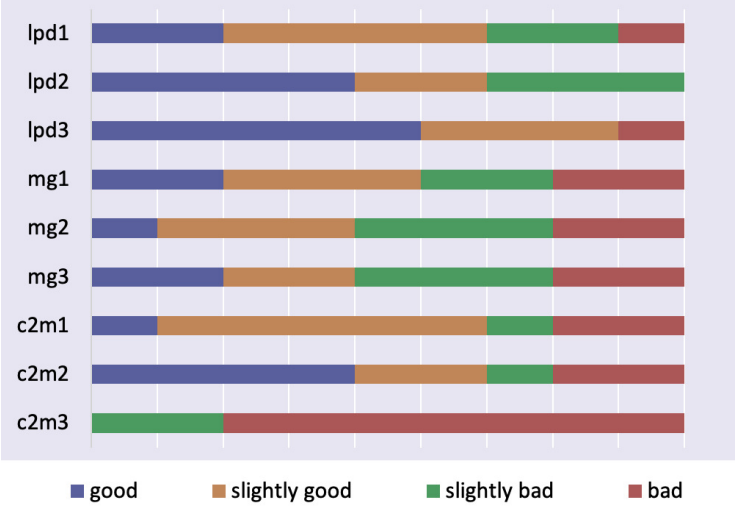


Figure 1: all results

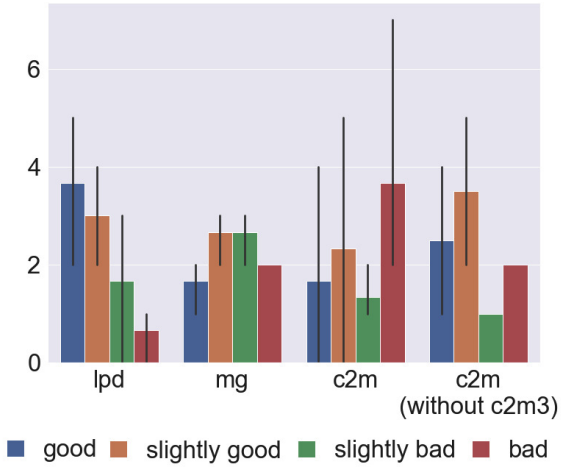


Figure 2: averaged results

5 Conclusion

We indicated that the neural networks using a transformer be able to generate music. GPT-2 model can produce better music than existing studies, even though it has not been optimized for music production. The music produced varies, and sometimes it produces music that is clearly bad. There is potential for improvement by tuning the values of the meta-parameters. Also, the model itself has room to be optimized for music generation.

A clear advantage of our implementation over the music generation in previous studies is that there is no limit to the length of the music to be generated. Our implementation is capable of generating very long melodies that have the integrity of a music. This is a major advance over existing studies that have only generated short pieces of music. The ability to change the tune of the generated melody by changing the value of the meta-parameter is a side benefit of our implementation.

5.1 Examples of generated music

Below is an example of the piano roll for the melody generated by Chord2Melody. The values of the meta-parameters are as shown in the figure. All have no external input (generated as a continuation of "<|endnote|>").

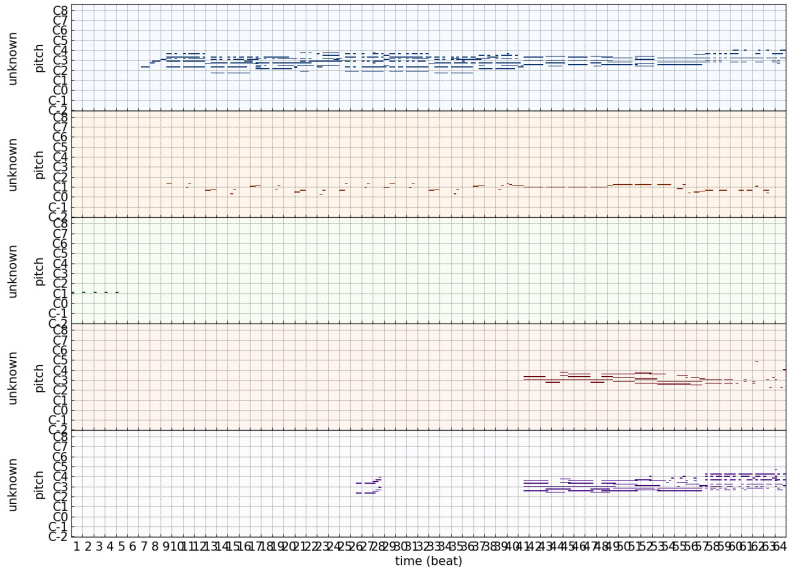


Figure 3: 5tracks,length=16bars,top_k=0,top_p=0

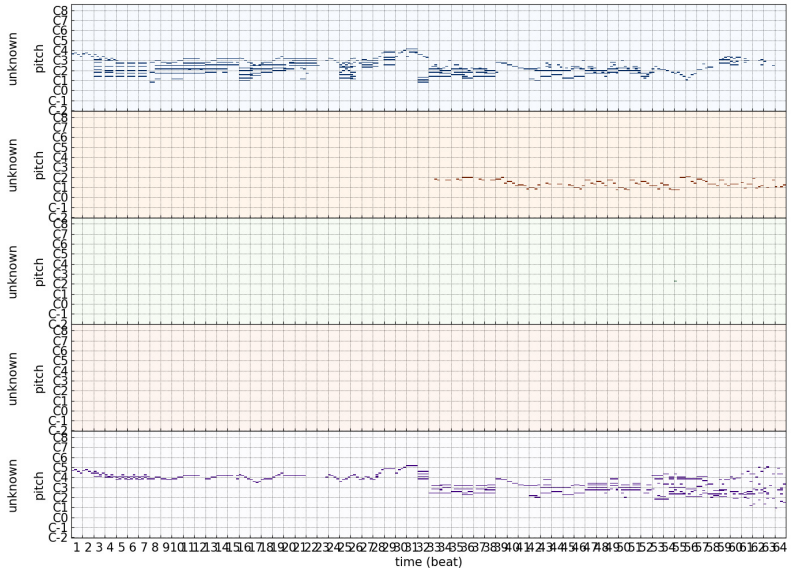


Figure 4: 5tracks,length=16bars,top_k=0,top_p=3

Figure 5: 5tracks,length=16bars,top_k=0,top_p=7

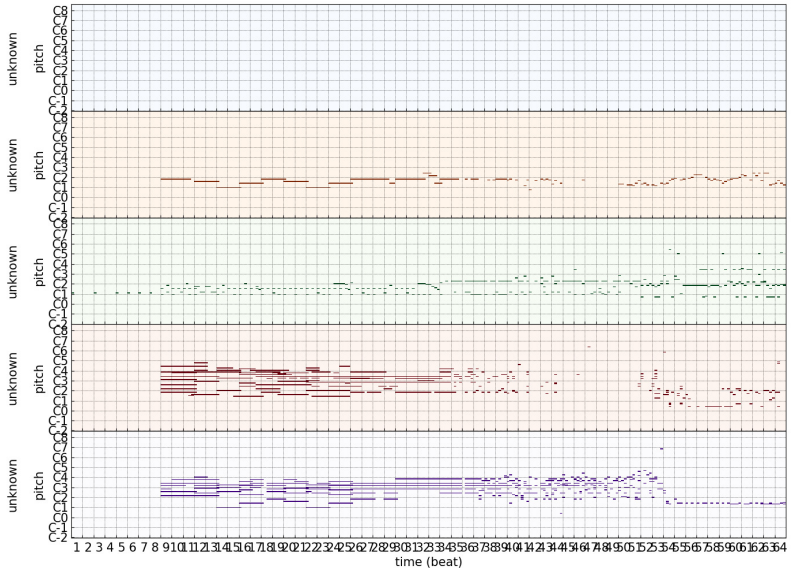


Figure 6: 5tracks,length=16bars,top_k=0,top_p=10

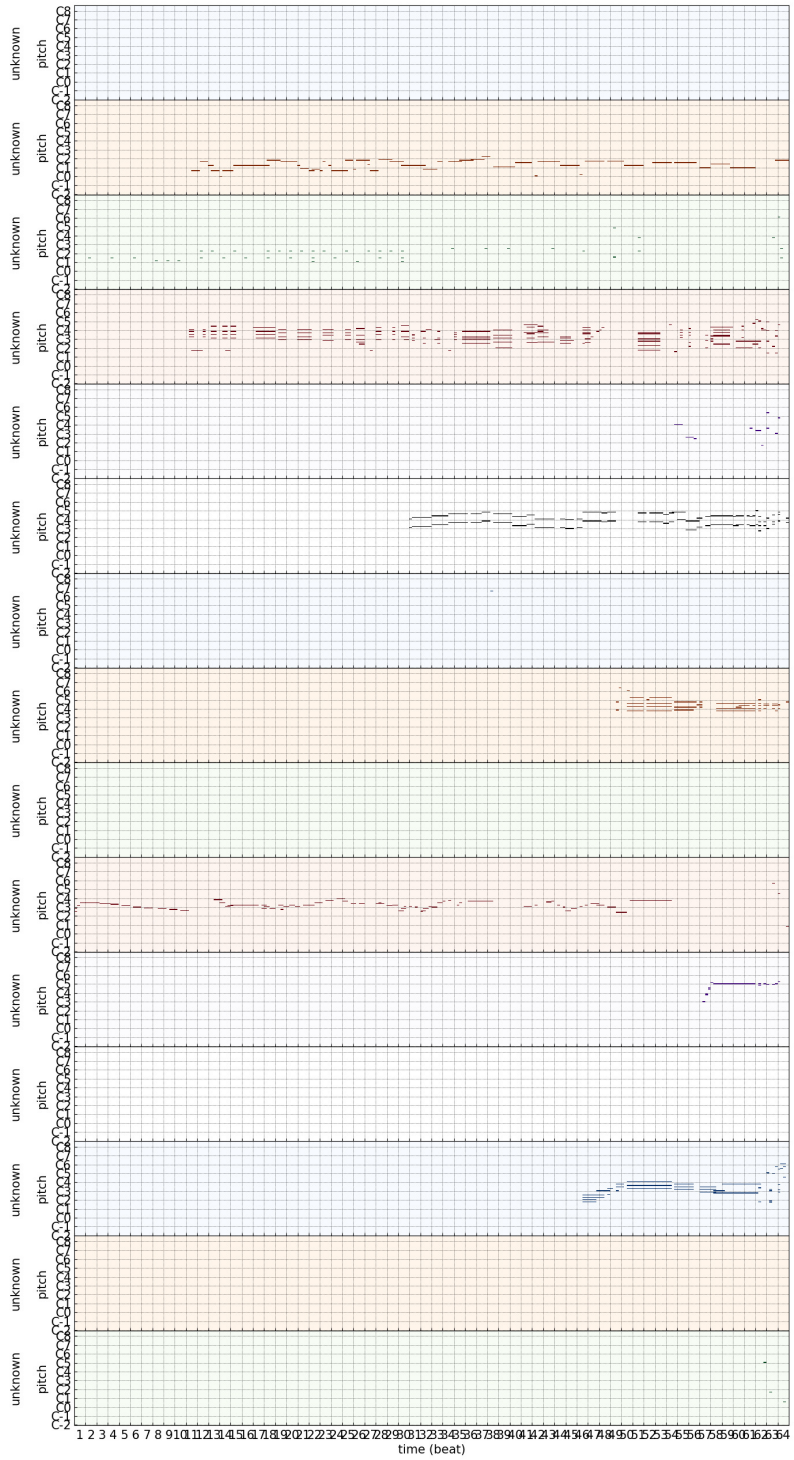


Figure 7: 17tracks,length=16bars,top_k=0,top_p=7

References

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. Improving language un- derstanding with unsupervised learning. Technical report, OpenAI. 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language under- standing. In *North American Association for Com- putational Linguistics (NAACL).* 2019.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197.* 2019.

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

Barz, B. and Denzler. J. Do we train on test data? purging cifar of near-duplicates. *arXiv preprint arXiv:1902.00423.* 2019.

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang and Yi-Hsuan Yang (equal contribution). MuseGAN: Demonstration of a Convolutional GAN Based Model for Generating Multi-track Piano-rolls. In *Late-Breaking Demos of the 18th International Society for Music Information Retrieval Conference (ISMIR). (two-page extended abstract)* 2017.

Chen, Mark and Radford, Alec and Child, Rewon and Wu, Jeff and Jun, Heewoo and Dhariwal, Prafulla and Luan, David and Sutskever, Ilya. Generative Pretraining from Pixels. 2020.

Colin Raffel. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. PhD thesis, Columbia University. 2016.