MusicMLM: A Resource Efficient Masked Language Model Fine-Tuned for Music Production

Jackson Argo `jacksonargo@berkeley.edu`

Abstract

With the rapid advances in transfer learning and release of large scale language models like BERT, it has become much easier to produce language models with a high level of accuracy in specific domains using a significantly smaller training set. This project aims fine-tune a pretrained model using a corpus of music production textbooks to produce a masked language model that can be used in the field of music production.

1 Introduction

Music production is a highly specialized field of its own, but requires knowledge across many other specialized domains. This ranges from computer science and engineering to music theory and design, and even project management. The broad range of topics covered in music production make it an interesting field of study for natural language processing and language models. Music production already relies heavily on computers and automation, and an effective language model could be easily integrated into existing production tool sets.

In this paper, I show that it is possible to produce a usable masked language model with a relatively fast training time and small corpus of text.

2 Background

The large-scale language model BERT, (Devlin et al., 2018), was released with the purpose that it could be taken and fine-tuned to produce domain specific language models. The tooling to produce these downstream language models has grown in both adoption and usability, however running the training algorithm can still take a considerable amount of time and computation resources (even though the BERT authors suggest otherwise).

This limitation served as the inspiration for DistilBERT, (Sanh et al., 2020), which offers the same features as BERT, but with a smaller resource footprint. According to the paper, DistilBERT is able to a the retain "97% of BERT performance," with only "40% of the size." DistilBERT was trained using a model compression algorithm introduced by Hinton et al. (2015) with BERT as the base model. The algorithm introduces a softmax temperature parameter to smooth the softmax prediction distribution, where $T = 1$ is the standard softmax function, and higher T produces a smooth prediction function. The *distilled* model is trained using the smoothed softmax and significantly fewer layers than the parent model, with the training goal that the new model produces the same prediction as the base model.

FinBERT, (Yang et al., 2020), is a domain language model trained using BERT for text-classification and regression using data from financial texts. FinBERT was evaluated against other language models using financial sentiment analysis text, and was shown to be very successful. This project similarly trained language models from other base models such as ELMo and ULMFit, and showed that FinBERT outperformed these models as well.

Lee et al. (2019) produced a domain specific language model for use in biomedicine. Due to the heavily specialized language used in biomedicine, the authors chose further pretrain BERT on a corpus of biomedical literature, before fine-tuning it for specific use-cases. These use-cases include named-entity recognition, question answering, and relation extraction. This project made minimal changes to the overall architecture of BERT, and focused primarily on leveraging the volume of domain specific literature that is available. BioBERT was trained on a fairly large corpus of data, relative to this project, and saw significant performance improvements, similar to FinBERT.

3 Methods

3.1 Model Selection

This project aims to strike a balance between performance, portability, and training time. Given the strong performance of downstream BERT models and the high performance and relatively fast training speed of DistilBERT, I chose DistilBERT as the base mode for training. The training goal is to fine-tune a masked language model from a corpus of music production literature.

3.2 Dataset

The data was collected from an assortment of college-level textbooks related to music production. These textbooks cover a wide range of topics from abstract discussions of the philosophy of music production to practical how-to guides and reference tables.

3.3 Preprocessing

The textbooks were provided as pdfs, so the first step was to strip the text from the pdf files. (Older textbooks that were image scans as opposed to properly typeset were discarded). In a similar fashion to the preprocessing used by BERT, I removed supplemental text outside of the main body of text; this included headers, captions, tables, etc. Captions often referred directly to information in a picture or table, which cannot be captured by the model, and table entries themselves are often not complete sentences or standalone ideas. I also removed introduction section, which primarily focused on upcoming content within the textbooks themselves, as opposed to providing standalone information. Preprocessing was largely manual due to the inconsistent formatting between textbooks. In some cases, I did chose to keep captions that were did present complete ideas or full sentences. After manually removing unneeded content, the text was still in paragraph form. I used a pretrained sentence tokenizer Punkt, (Kiss and Strunk, 2006), to split the corpus into sentences. Finally, I removed one word sentences and control characters from the data.

As I was developing the model and training scripts, I quickly realized that the more time I spent combing through the corpus and deleting improperly tokenized sentences, the better and faster training results could be. After several manual QA passes through the training corpus, I was able to train the models in a reasonable time with decent results for this project. It is very likely that by spending more time cleaning and prepping the dataset, you could find even better training results.

3.4 Training

3.4.1 Setup

I chose to fully randomize the sentence order of the full dataset so that splitting the dataset for training and benchmark is trivial. The order of BERT's training sentences are not fully randomly randomized, which allows it to make next sentence predictions, among other things, however the downstream model does not need this feature.

The sentences were split into a three datasets: a training dataset, an evaluation dataset used during training, and a benchmark dataset fully excluded from the training process. The benchmark dataset is used to evaluate the downstream model and the BERT and DistilBERT base models.

To tokenize the sentences into words, I used the pretrained BERT tokenizer provided by transformers python library. In order to optimize training speed, I chose to limit the sentence length to 64 tokens and fully padded all sentences. (This is the same length used by FinBERT.)

Dataset	# of Sentences
Training	63,738
Evaluation	21,247
Benchmark	21,246

3.4.2 Parameters

The appendix of the BERT paper gives fairly detailed instructions for hyper-parameter tuning. Following the recommendation, I trained the model with the following fixed parameters: attention heads = 12, layers = 6, dropout = 0.1, and activation=gelu. The training procedure uses the same masking rate as base BERT and masks 15% of the input tokens, of which 80% of are masked using the mask token, 10% using a random word, 10% remained unchanged. Following the recommendation from the BERT paper, I trained multiple models using AdamW optimizer, (Kingma and Ba, 2017) and different combinations of batch size, learning rate, and epochs.

3.5 Evaluation Loss

I used the pytorch and transformers python api's to perform the training. (The training source code can be found here.)

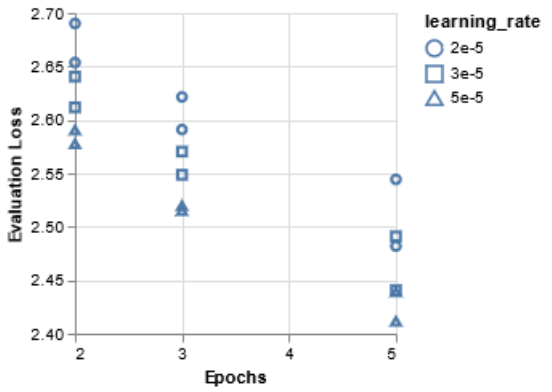

Figure 1: Evaluation loss measured during hyperparameter search.

The top performing models used the smallest learning rate and most number of epochs. Decreasing the learning rate and increasing number of epochs typically resulted in better evaluation for this model. Batch size seemed to have a small effect overall. Figure 1 shows the evaluation loss for each learning rate and number of epochs.

The optimum configuration found in training was batch size = 16, learning rate = 5e-5, and epochs = 5.

4 Results

4.1 Benchmark

The benchmark dataset is masked with the same ratio used in training and used to measured the evaluation loss of BERT, DistilBERT, and MusicMLM. MusicMLM saw a significant 36% improvement in evaluation loss compared to the base models.

	Evaluation Loss
BERT	3.73
DistilBERT	3.78
MusicMLM	2.42

Table 1: Evaluation loss from benchmark dataset.

4.2 Practical Examples

The goal of the project is to produce a language model that can be used in music production, so in additional to an automated evaluation, I also chose to evaluate Bert, Distilbert, and MusicMLM using more practical use-cases of masked language models. (You can run your own unmasking tasks using MusicMLM here.)

4.2.1 Unmasking

The first experiment consists of simple word unmasking with music related sentences. The sentences are purposefully missing context and can have many valid answers. For each sentence, the prediction from each of the models are different, but valid, and MusicMLM gives the most opinionated prediction.

BERT	piano (0.111)
DistilBERT	accordion (0.088)
MusicMLM	guitar (0.138)

Table 2: Input: *The best instrument for recording is [MASK].*

BERT	different (0.124)
DistilBERT	louder (0.110)
MusicMLM	better (0.0.092)

Table 3: Input: *Increasing the gain produces a [MASK] sound.*

4.2.2 Analogies

A powerful feature of language models is the ability process analogies, (Brown et al., 2020). Word embeddings can be used with simple addition and subtraction to provide reasonable predictions for analogies. In order to be consistent with the previous test and to better stretch the limits of these models, I chose not use the arithmetic predictions, and instead use a masked sentence in the form *X is to Y as Z is [MASK]*. The predicted answer is the word with the highest score, minus any words used in the analogy. Unlike the previous experiment, the analogies have very distinct answers. Both Bert and MusicMLM predict the correct answer for the first analogy, shown in Table 4, but none of the models where able correctly answer the second analogy, Table 5, which uses more domain specific wording.

BERT	brass (0.253)
DistilBERT	sing (0.111)
MusicMLM	brass (0.168)

Table 4: Input: *Flute is to woodwind as Trumpet is to [MASK].*, Answer: *Brass*.

4.2.3 Bias

A well-known limitation of language models is the bias present in word embeddings, (Bolukbasi

BERT	pulses (0.153)
DistilBERT	transmit (0.060)
MusicMLM	perceive (0.045)

Table 5: Input: *Intensity is to decibels as frequency is to [MASK].*, Answer: *Hertz*.

et al., 2016). To evaluate the bias in MusicMLM, I looked at the top 3 predicted occupations for a man, woman, and person. MusicMLM's max prediction in all three cases is producer, and in each case there is a relatively large difference between the scores for the first and second predictions. The prediction scores for the top three words from BERT and DistilBERT are much more evenly distributed.

BERT	carpenter, waiter, barber
DistilBERT	carpenter, blacksmith, tailor
MusicMLM	producer, cop, slave

Table 6: Input: *The man worked as a [MASK].*

BERT	nurse, waitress, maid
DistilBERT	nurse, maid, waitress
MusicMLM	producer, librarian, supervisor

Table 7: Input: *The woman worked as a [MASK].*

BERT	farmer, teacher, nurse
DistilBERT	carpenter, farmer, clerk
MusicMLM	producer, mixer, supervisor

Table 8: Input: *The person worked as a [MASK].*

5 Conclusion

This project trained a masked language model called MusicMLM by fine-tuning the pretrained language model DistilBERT. The model saw a significant improvement in loss over both DistilBERT and BERT when evaluated using a domain specific corpus. This shows that the performance trade-off when using a compressed version of BERT can be overcome by fine-tuning. Additionally, MusicMLM showed improvements over the base models in several practical use cases.

5.1 Future Work

Due to the flexibility of transfer learning models and it's light-weight size, MusicMLM can be quickly extended with additional training tasks such a question and answer or next sentence prediction. These would be powerful additions to the model and broaden it's general usability. Additionally, as shown by other transfer learning language models like FinBERT and BioBERT, MusicMLM could also benefit from further pretraining before fine-tuning for specific tasks, to produce an even better performing language model.

References

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications.