Transformer-Based Music Language Modelling and Transcription

Christos Zonios
chzonios@cs.uoi.gr
Dept. of Computer Science and Engineering, University of Ioannina
Greece

John Pavlopoulos
annis@aueb.gr
Dept. of Informatics, Athens University of Economics and Business
Greece

Aristidis Likas
arly@cs.uoi.gr
Dept. of Computer Science and Engineering, University of Ioannina
Greece

ABSTRACT

Automatic Music Transcription (AMT) is the process of extracting information from audio into some form of music notation. This challenging task requires significant prior knowledge and understanding of music language. In this paper, we examine Transformer-based approaches for performing AMT on piano recordings by learning music language representations. We propose a new Music Language Modelling (MusicLM) pre-training approach for Transformers. It is based on an appropriately defined transcription error-correction task, and enables transfer learning for various musical tasks. Furthermore, a novel model for AMT is proposed that appropriately exploits a BERT Transformer for the MusicLM problem, showing the potential of transfer learning from Natural Language to MusicLM. We apply the Transformer on a Masked MusicLM task, and achieve musically coherent results. We also replace the RNNs used in current AMT models with pre-trained BERT-based Transformers, achieving improvements in AUC.

CCS CONCEPTS

• **Information systems → Content analysis and feature selection**; • **Computing methodologies → Neural networks**; *Transfer learning*.

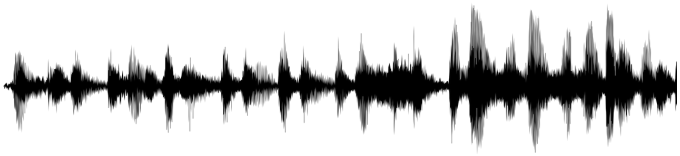
KEYWORDS

automatic music transcription, music language modelling, transformers, deep learning

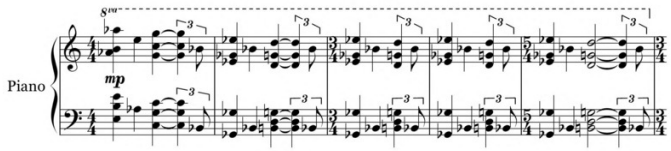
ACM Reference Format:
Christos Zonios, John Pavlopoulos, and Aristidis Likas. 2022. Transformer-Based Music Language Modelling and Transcription. In *12th Hellenic Conference on Artificial Intelligence (SETN 2022), September 7–9, 2022, Corfu, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3549737.3549754

1 INTRODUCTION

Automatic Music Transcription (AMT) concerns the process of automatically converting an audio signal to a high-level representation of the musical information present in it. When musicians perform transcription of music, they listen to the audio and use some form of music notation to generate a human-readable representation of that audio (Figure 1). Another musician can then use this representation to perform the music, by interpreting this notation. A subfield of Music Information Retrieval (MIR), AMT has been studied extensively [5, 6, 9, 10, 14, 18–20] due to its applications in musical analysis, teaching of music, annotation and others [6].



(a) Audio wave



(b) Sheet music

Figure 1: Transcription involves the extraction of musical information from sound (a) and its transfer to human-readable music notation (b).

Music Language Modelling [18], dubbed MusicLM, can be seen as the musical equivalent of language modelling in Natural Language Processing (NLP). By modelling the language of music, and specifically its temporal, melodic, rhythmic and harmonic structure, as well as emergent patterns and repeated passages, we can not only obtain better understanding, but also create better representations and abstractions. This is an essential step towards solving various MIR problems while in the context of AMT, it can allow the prediction of more realistic transcriptions [14], improving transcription accuracy and increasing the confidence of model predictions.

Transformer architectures [22] have introduced great improvements in a variety of sequence modelling problems, showing robustness in each particular application and transfer learning potentials. This is done by fine-tuning pre-trained models to achieve state-of-the-art performance in NLP and Speech Recognition applications, using only a fraction of resources required by previous approaches, such as available data, time and computational power [2, 7, 8].

The primary motivation for this work is *the study and use of Transformer-based NLP language models in an end-to-end AMT network*. Considering the parallels between natural and music language, our intuition is that Transformers can replace RNN-based

SETN 2022, September 7–9, 2022, Corfu, Greece

ACM ISBN 978-1-4503-9597-7/22/09…$15.00
https://doi.org/10.1145/3549737.3549754

models currently used for AMT. Besides possible performance benefits, Transformers are advantageous over previous sequence modelling architectures due to their attention mechanism that can be used to visualise the sequence elements that attribute most to the prediction [23]. This in turn can provide valuable insights regarding the model's decision process [1] and potentially lead to explainable models. The main contributions of this work are the following:

(1) We investigate the potential of transfer learning from natural language to tasks related to music, by evaluating BERT [8] for the task of Music Language Modelling.
(2) We propose a novel musical note denoising objective, which can be used as music transcription error detection/correction task for pre-training.
(3) We propose a novel Transformer-based AMT model that surpasses in AUC the baseline RNN models.

The rest of this paper is organized as follows: Section 2 introduces the current RNN-based architectures for AMT as well as existing Transformer architectures used in musical tasks. Section 3 introduces the dataset used in this work and details our proposed approaches, while in Section 4 we explain our experimental setup. Finally, in Section 5 we present the experimental results of this study, and in Section 6 we provide conclusions and future work directions.

2 RELATED WORK

In the field of AMT, most current architectures use Recurrent Neural Networks (RNNs) for modelling musical sequences. Hawthorne et al. [10] introduced the Onsets and Frames (*O&F*) model, which jointly learns to predict note activations as well as note onsets. It has one convolutional stack followed by a bidirectional Long-Short Term Memory (BiLSTM) [17] per task, and then uses the joint outputs to make the final frame activation prediction using an additional BiLSTM layer, as can be seen in Figure 2 (a). They followed up this work with [12], which introduced a few modifications, most notably an offset detector stack and a larger embedding dimension, as well as a new dataset for AMT training and evaluation. Another noteworthy work was published by Kim and Bello [14], who extended the *O&F* architecture by using a Generative Adversarial Network step after the prediction was acquired, in order to learn to output more realistic predictions.

The Music Transformer [13] is a Transformer model trained on the task of harmonizing a melody, which entails predicting the notes sang by the bass, tenor and alto given the notes sang by the soprano. It uses a sparse, MIDI-like representation of music, and employs relative attention. Wave2Midi2Wave [12] builds on the Music Transformer by introducing a WaveNet model for generating audio from the symbolic representations. It also uses the *O&F* transcription model to work with raw audio as its input, and shows that an AMT model can be used to provide music representations for pre-training a Transformer by predicting future notes. Recently, Hawthorne et al. [11] proposed a novel Transformer-based model for AMT, adapted from the Music Transformer, which achieved new state-of-the-art results. In this work, we consider an unmodified, pre-trained NLP BERT Transformer as our Music Language Model, and use it as a drop-in replacement for RNNs.

3 MUSIC LANGUAGE MODELLING

In this section, we introduce the dataset used in this work and describe our contributions and methodology for pre-training and evaluating a BERT Transformer for MusicLM tasks. Finally, we describe how we integrate it into our AMT model.

3.1 MAESTRO Dataset

The MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) [12] is a music dataset containing about 200 hours of piano performance audio recording and MIDI pairs, finely aligned. We use the specified train/validation/test split provided by the authors, which contain 96.3/11.8/12.1GB of performances respectively. More details regarding the dataset files are provided in Appendix A.

In order to produce the frame labels, we use a piano roll representation. The piano roll is a tensor of shape $(N_{frames} \times 88)$, where each frame corresponds to r input audio samples in its receptive field and contains the note information for that time frame. Each frame is a vector of size 88, with each element being an integer (1 for onset, 2 for inside, 3 for offset) as in *O&F*. Each frame's receptive field is spaced by `hop_length` samples from the next frame.

The input to the AMT models is a vector of $N_{frames} \cdot r$ audio samples. A preprocessing step is always performed to produce a mel-spectrogram representation of the input, using the same parameters as in the *O&F* model.

3.2 Evaluating BERT for Music Language Modelling

We introduce a novel text representation of the input frames, and define a Masked Music Language Modelling (MMLM) task, similar to the Masked Language Modelling task used in NLP (hereafter MNLM). We then describe how we can bypass the tokenization step and pass the input frames directly as embeddings, in order to make the model trainable end-to-end.

In order to use BERT for MusicLM, we transform each frame into a string literal in order to train and use a standard text tokenizer on the music representations. That is, each integer in a frame is turned into a character. A sentence is then a sequence of string representations of frames, separated by a whitespace character.

Training a tokenizer on a dataset of such sentences yields words (string representations of frames), subwords (strings representing patterns of active and inactive notes, for example "`10001`" to represent a major third interval) and special tokens such as the start and end of a sentence, unknown and mask tokens.

A problem we encountered is that the language of piano music, unlike natural languages, such as English, has a vocabulary too large to fit in memory (2^{88} possible words or frame states). However, the overwhelming majority of these states are extremely unlikely to occur (for example, states with more than 10 notes or with notes with large distances between them) and do not show up in datasets.

Including the musical pause, few tokens are present in most of the training dataset. The most common tokens after the musical pause are those that correspond to only one note being played. This means that the dataset is very sparse, with orders of magnitude more negative (note off) than positive (note on) samples per frame. The total number of unique tokens in the training set is 885,768.

For our purposes, we trained a BERT (base) model with an MNLM head by using only a subset of the vocabulary consisting of the N-most frequent tokens. We tried $N \in \{2{,}000, 10{,}000, 50{,}000\}$ but found no difference in the performance of the network. Thus, we opted for the lowest number to speed up the training procedure.

Preliminary evaluation on masked predictions on unseen music sequences showed that both model performance in the per-frame F_1 score and the subjective measures of listening for how realistic the predicted sequences are, offer promise that the model is well suited for MusicLM. The problem with this approach is that for an AMT model, the string encoding step is not differentiable as a thresholding step is required between the continuous outputs of the activation estimator and a language model. Although there exist approaches to make the process differentiable, such as in [3], we consider a simpler solution. We remove string encoding and tokenization altogether and instead pass the input frames as embeddings directly, bypassing BERT's internal learned lookup tables that transform the input token IDs into embedding vectors. This approach worked well and within a few epochs the model achieved an F_1 score higher than 90%.

3.3 Training BERT for MusicLM

We introduce an appropriately defined pre-training task for MusicLM based on transcription error correction. We extend the *O&F* model with a BERT Transformer pre-trained on this task in order to improve its transcription accuracy by detecting and fixing transcription errors.

Notes typically span multiple time frames and exhibit overlap, while sentences are not clearly defined in music. Hence, MNLM and next sentence prediction (NSP) may be suitable objectives for NLP tasks, but not for AMT. This observation motivated our investigation for the development of a BERT pre-training objective suitable for AMT. We have considered a note denoising objective, by introducing noise in the input that imitates transcription errors. To our knowledge, this is the first attempt at creating a music transcription error detection/correction objective.

The addition of noise is made per frame. Each active note has a p_1 chance of becoming inactive, and each inactive note has a p_2 chance of becoming active. We empirically set these values to 50% and 10% respectively, by following the typical transcription error distribution. We have observed that False Negatives are more frequent, because a typical classification error occurs when some of the simultaneously active notes are missing. False positives may also occur when other similar notes are activated. This may occur, for example, when a note contains harmonics of the same frequency as the False Positive. Negative samples are also vastly more common than positive samples.

Our BERT decoder consists of a fully connected layer added on top of BERT base, which outputs a sequence of frames. No other BERT pre-training objective was used besides our denoising one. At the base of the network, another fully-connected layer is added in order to transform the input noisy frames into embeddings of the same size as the ones that BERT uses internally.

We integrated the above transcription error correction module as an extra module on top of the *O&F* model, which is used as an encoder (encoding audio into a music representation). For the purpose of evaluating the efficacy of the model, we completely freeze the weights of the encoder and only fine-tune the decoder. However, since the inputs of the decoder match the outputs of the encoder, this model could also be fine-tuned or re-trained end-to-end. We call the resulting pre-trained decoder module as *TEC-BERT*. The architecture of the model is shown in Figure 2 (b).

3.4 The Orpheus model for AMT

We propose a novel model based on the *O&F* architecture, for AMT. We keep the pre-processing of the raw audio into a mel-spectrogram [21], replace *O&F*'s BiLSTM layers with BERT Transformers in an attempt to improve the music language modelling part of the model, and adjust the architecture so that the model does not get too large for our experimental system (Appendix B). We will hereafter refer to this model as *Orpheus*. The architecture of the Orpheus model is presented in Figure 2 (c). The differences between our model and the *O&F* model are outlined as follows:

- We use a common convolutional stack (ConvStack) and pre-trained BERT encoder for the Onset and Offset predictors, whereas the *O&F* model uses two separately trained ConvStack and RNN encoders.
- We introduce a fully connected (Linear) layer that transforms the outputs of the Onset and Offset predictors and the Activation estimator into the appropriate embedding dimension for the BERT decoder.
- We replace the RNN decoder with a pre-trained BERT decoder before the the final layer of the model.

4 EXPERIMENTS

4.1 Performance Metrics

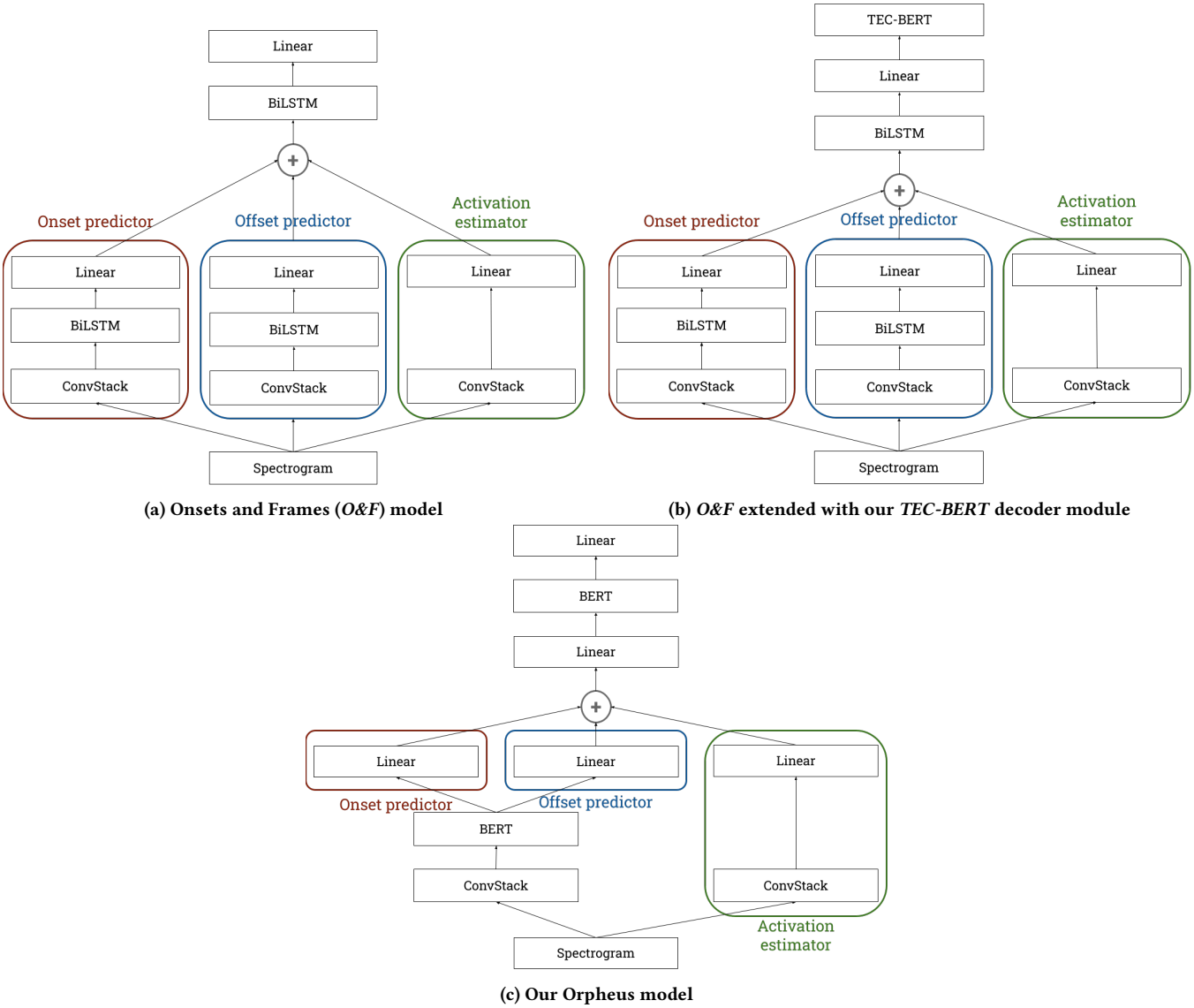
The metrics used to evaluate AMT model performance are the same as in [10], first described by Salamon et al. [16], with the addition of an Area Under the receiver operating characteristic Curve (AUC) metric, which unlike the F_1 score is classification threshold-agnostic. We present below all the metrics used:

- Frame Precision/Recall/F_1: a true positive is a note that was predicted correctly as active at that time frame;
- Note Precision/Recall/F_1: a true positive is a note whose offset is detected within $\pm 50ms$ of the ground truth and its frequency is found within 50 cents of the ground truth;
- Note with offset Precision/Recall/F_1: same as the note metrics, but the note also has to have an offset value within 20% or $\pm 50ms$ of the ground truth, whichever is larger;
- AUC: integral of the ROC curve, where a true positive is calculated as in the per frame metrics.

4.2 Experiment Setup

As a baseline, we trained a complete *O&F* model, extended to introduce the improvements and modifications described by the authors' follow-up work [12], for 640,000 steps (mini-batches) in our own setup. The baseline architecture is shown in in Figure 2 (a).

Both the proposed model *Orpheus*, the baseline and the baseline extended with a *TEC-BERT* module are trained for the same number of steps, although our own models are trained for a lower number of total epochs due to the smaller batch sizes compared to the baseline,



(a) Onsets and Frames (*O&F*) model

(b) *O&F* extended with our *TEC-BERT* decoder module

(c) Our Orpheus model

Figure 2: Comparison of the Onsets and Frames (*O&F*) model used as our baseline, *O&F* extended with our *TEC-BERT* decoder module, and our Orpheus model.

as they are much larger. Appendix B contains the specifications of the system used to conduct the experiments.

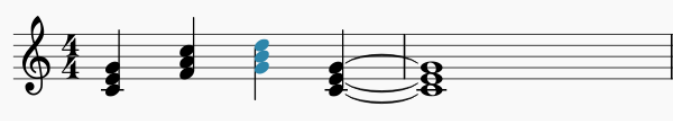
During experimentation, we found that most models were very sensitive to the note-on threshold th_{note}, which is the threshold of a predicted probability resulting in a note being classified as active. Models with Transformer decoders seemed particularly sensitive to this threshold. We also define an onset threshold th_{onset} that also affects performance.

We tuned the thresholds per model on a validation set. As th_{note} and $th_{onset} \in (0, 1)$, we begin at 0.01 and increment by 0.01 until we reach 0.99. We keep th_{onset} fixed for a whole pass over the range for th_{note}, then increment th_{onset} and pass over the whole range of th_{note} again, and repeat until all combinations of threshold values are covered. The best score observed on the validation set is used to select the optimum thresholds per model.

5 EXPERIMENTAL RESULTS

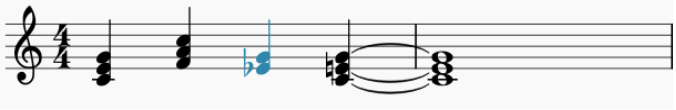
5.1 Music Language Modelling

Prior to assessing AMT performance (Sec. 5.2), an empirical evaluation was conducted to provide a subjective assessment of BERT for MusicLM. Our experiments show that BERT trained with a MNLM task adequately learns to predict musical sequences, even generating subjectively more musically pleasing and interesting sequences than the ones artificially created for the experiments.



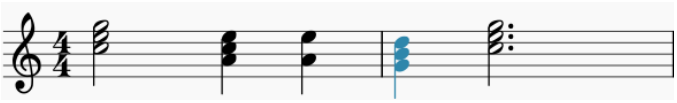
(a) Ground truth (masked notes in blue).

(b) Topmost prediction: same as the chord that follows (imperfect fall). Presumably, this is the top prediction because it puts the I chord in the key of C in a strong position in the measure.

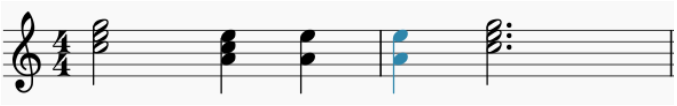


(c) Second prediction: E♭ and G, inferred as a C minor (borrowed) chord in the key of C major, or a suspended dominant chord (V, VII) in the key of E minor that is solved into the VI of the scale.

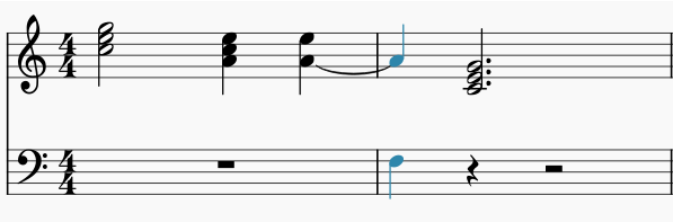
Figure 3: The two top MMLM predictions (b,c) on masking the V chord of a perfect fall in the key of C major (a).



(a) Ground truth (masked notes in blue).



(b) Topmost prediction: repeat/continuation of previous chord, the VI of the C major scale with the subtraction of the third. It can also be inferred as the I in the key of A natural minor, as it appears in a strong position of the measure.



(c) Second prediction: IV or II which are subdominant chords of the C major scale (imperfect fall).

Figure 4: The two top MMLM predictions (b,c) on a musical sequence in the key of C major (a).

We observed that the BERT Transformer model trained on the Masked Language Modelling task produced believable musical sequences on the MMLM task. Figures 3 and 4 provide illustrative results where although MMLM is not given any information about the key, it predicts chords that adequately fit the context. In Figure 3, MMLM suggests an imperfect fall (b), putting the I chord in the key of C in a strong position in the measure. This is fairly standard in music as the I chord usually sits in a strong position in the measure, and imperfect falls are far more common than perfect falls. Interestingly, the second prediction is a chord that does not belong to the key of C. A different context was assumed by the model, producing a far more interesting and auditorily pleasing result with the E♭ note resolving up to E natural. This can be explained by music harmony as either a C minor borrowed chord in the key of C major, or as a suspended dominant chord in the key of E minor. In Figure 4, MMLM suggests either a continuation of the previous chord (b) or an imperfect fall (c). The aforementioned findings show MMLM's ability to predict chords that are justifiable with music harmony, even in very small sequences, reflecting the model's ability to infer musical context.

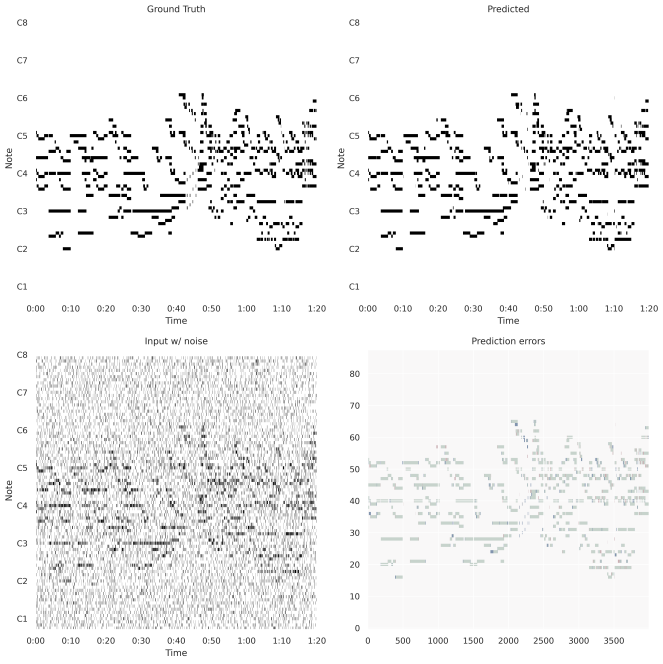


Figure 5: Transformer further pre-training results. Using a noisy input, our model learns to correct the errors and output a more realistic transcription. Green, light green, red and blue in the lower right picture represent TPs, TNs, FPs, and FNs respectively. There are very few FPs.

On the noisy input training task, the *TEC-BERT* model produces clean predictions on the test set. Figure 5 presents a visualization of the noisy input, prediction and ground truth. It can be observed that the model learns to output realistic transcriptions and makes few mistakes given a noisy input.

When initialized with the base (natural language) uncased pretrained checkpoint, BERT learned and performed better than a randomly initialized version on the task of denoising music input. This might mean that there is at least some transfer learning potential from NLP to MusicLM. Table 1 shows the test set results of an NLP pre-trained model versus a randomly-initialized model. The per-note metrics are lower because of the high probability of

setting the onset to zero, making it very hard for the model to detect it in the allotted frames.

5.2 Automatic Music Transcription

Table 2 shows the experimental results for AMT while Figure 6 presents a comparison between the predictions of the baseline model (*O&F*), *O&F* extended with our *TEC-BERT* decoder module, and our *Orpheus* model. Illustrative visualizations of predicted probabilities and errors can be found in Appendix C. By extending the *O&F* model with a *TEC-BERT* decoder, we find that no additional benefit is gained over the threshold-tuned baseline model. However, its predictions are more confident compared to the ones of the baseline, as can be observed in Figure 6.

It can be observed that *Orpheus* achieves the best AUC. However, it achieves a lower Precision and Recall. This hints that it is able to provide better transcription probabilities than the baseline, but is more sensitive to thresholding, and may benefit from a different thresholding approach.

The addition of our *TEC-BERT* decoder appears to provide comparable performance with the baseline model, while having the advantage of being able to be pretrained separately. Moreover, possible advances in the pretraining tasks and Transformer architectures for MusicLM might directly translate to increase in AMT performance.

6 CONCLUSIONS

We have investigated and shown that a BERT model, pre-trained for natural language tasks, is suitable as a fine-tuning starting point for music language modelling. Also, there is evidence of transfer learning potential between natural and music language.

When replacing Bidirectional Long-Short Term Memory layers in a current *O&F* AMT model with a BERT-based Transformer, the resulting (*Orpheus*) model is able to learn contextual representations and achieves a transcription accuracy on par with the state of the art.

More specifically, we were able to approach in all metrics, and surpass in AUC, the RNN-based baseline. We consider this an advancement that enables the transfer learning and explainability benefits of Transformers for Music Information Retrieval and Music Language Modelling problems.

Future work will comprise more Transformer architectures, such as T5 [15] used in [11], or the Longformer [4] which can model significantly longer sequences, to assess their music language understanding capabilities. The proposed decoder pre-training approach could be augmented by appropriately designing input noise to more closely mimic transcription errors. As Transformers are particularly sensitive to note-on thresholds, we expect that more intelligent thresholding techniques may provide more robust models in the future. This might be achieved by jointly training a thresholding submodule, or using additional losses and techniques to force the network to predict probabilities closer to 0 and 1. Finally, a novel music notation tokenization method, perhaps based on the representations used in the Music Transformer but compatible with absolute positional embeddings, would enable pre-training NLP models such as BERT without compromises. Such a method is, in our opinion, the key to unlocking the power of Transformer architectures and enabling new approaches for various MusicLM tasks.

REFERENCES

[1] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2020. Visbert: Hidden-state visualizations for transformers. In *Companion Proceedings of the Web Conference 2020*. 207–211.

[2] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477* (2020).

[3] Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQˆ3: Differentiable Sequence-to-Sequence-to-Sequence Autoencoder for Unsupervised Abstractive Sentence Compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 673–681. https://doi.org/10.18653/v1/N19-1071

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[5] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. 2018. Automatic music transcription: An overview. *IEEE Signal Processing Magazine* 36, 1 (2018), 20–30.

[6] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. 2013. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems* 41, 3 (2013), 407–434.

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Johnathon Michael Ender. 2018. *Neural Networks for Automatic Polyphonic Piano Music Transcription*. University of Colorado Colorado Springs.

[10] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. 2017. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153* (2017).

[11] Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel. 2021. Sequence-to-sequence piano transcription with transformers. *arXiv preprint arXiv:2107.09142* (2021).

[12] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2018. Enabling factorized piano music modeling and generation with the MAESTRO dataset. *arXiv preprint arXiv:1810.12247* (2018).

[13] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music transformer. *arXiv preprint arXiv:1809.04281* (2018).

[14] Jong Wook Kim and Juan Pablo Bello. 2019. Adversarial learning for improved onsets and frames music transcription. *arXiv preprint arXiv:1906.08512* (2019).

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[16] Justin J Salamon et al. 2013. *Melody extraction from polyphonic music signals*. Ph. D. Dissertation. Universitat Pompeu Fabra.

[17] M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681. https://doi.org/10.1109/78.650093

[18] Siddharth Sigtia, Emmanouil Benetos, Srikanth Cherla, Tillman Weyde, A Garcez, and Simon Dixon. 2014. RNN-based music language models for improving automatic music transcription. (2014).

[19] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. 2016. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 5 (2016), 927–939.

[20] Jonathan Sleep. 2017. Automatic music transcription with convolutional neural networks using intuitive filter shapes. (2017).

[21] Stanley S Stevens and John Volkmann. 1940. The relation of pitch to frequency: A revised scale. *The American Journal of Psychology* 53, 3 (1940), 329–353.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

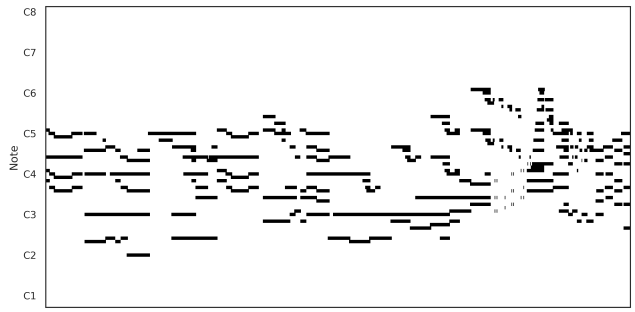
[23] Jesse Vig. 2019. A Multiscale Visualization of Attention in the Transformer Model. arXiv:1906.05714 [cs.HC]

Table 1: BERT performance on the music denoising task with different initializations. The NLP initialization refers to the pre-trained NLP BERT model (specifically, the `base-uncased` checkpoint).

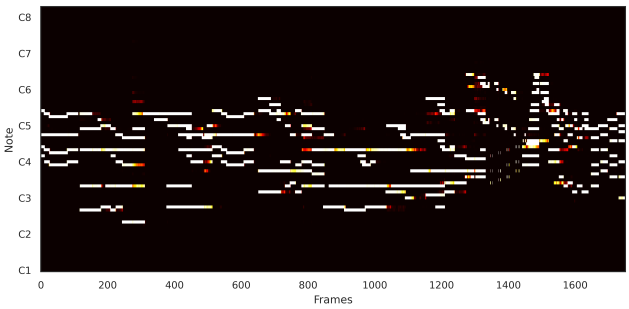
	Frame metrics			Note metrics		
Initialization	Precision	Recall	F1	Precision	Recall	F1
NLP	**93.32 ± 2.1**	**88.83 ± 3.2**	**91.01 ± 2.6**	**53.83 ± 3.7**	**39.84 ± 5.3**	**45.58 ± 4.1**
Random	91.42 ± 2.5	73.80 ± 5.7	81.60 ± 4.4	12.96 ± 2.8	17.86 ± 3.0	14.88 ± 2.5

Table 2: Model performance on the MAESTRO test set, using Precision (P), Recall (R), F1 and AUC. The top two rows comprise results taken from the respective published studies, where no AUC was reported.

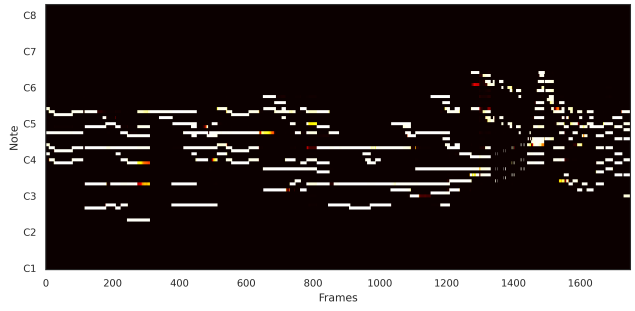
	Frame			Note			Note w/ offset			
Model	P	R	F1	P	R	F1	P	R	F1	AUC
O&F [12]	92.11	88.41	90.15	98.27	92.61	95.32	82.95	78.24	80.50	-
Kim and Bello [14]	93.1	**89.8**	**91.4**	98.1	**93.2**	**95.6**	83.5	**79.3**	**81.3**	-
Reproduction of *O&F*	92.45	88.71	90.50	97.41	92.76	95.02	83.16	79.23	81.13	99.30
O&F w/ *TEC-BERT* head	92.27	88.73	90.42	97.44	92.78	95.04	82.93	79.00	80.91	99.02
Orpheus	92.01	83.32	87.38	97.49	86.64	91.71	79.78	70.95	75.08	**99.38**



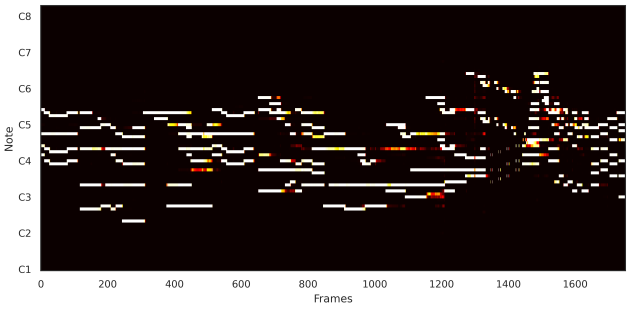
(a) Ground truth piano roll



(b) Prediction of the *O&F* model



(c) Prediction of the *O&F* model extended with our *TEC-BERT* decoder module



(d) Prediction of our proposed Orpheus model

Figure 6: Comparison of *O&F*, *O&F* + TECBert, and Orpheus model predictions

A DATASET AND PREPROCESSING

A.0.1 Audio Files. The audio files are in raw WAVE format. The `librosa` library was used to load the audio from the files, using a sample rate of $16kHz$.

A.0.2 MIDI Files. MIDI files contain musical information describing musical events such as a note turning on at a certain time with a certain velocity. We use the `pretty_midi` library in order to parse the musical information into a piano roll representation as described in Section 4.

B SYSTEM SPECIFICATIONS

The hardware system used to conduct all experiments reported in this work was the following:

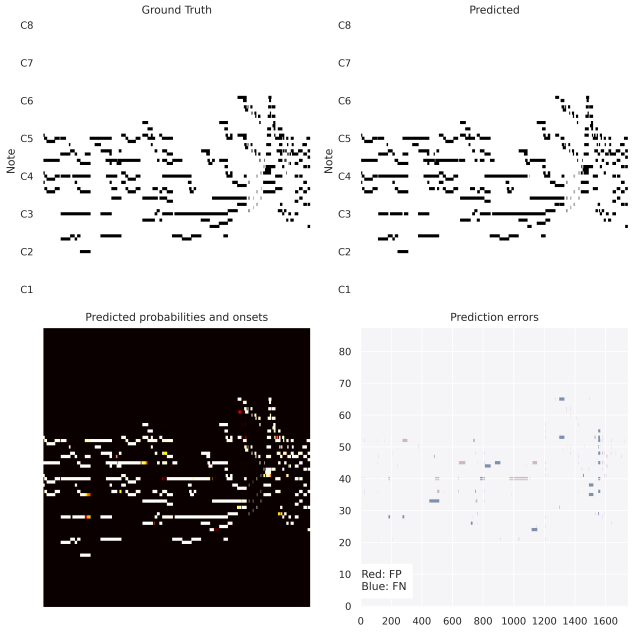


Figure 8: O&F model with a *TEC-BERT* head predictions

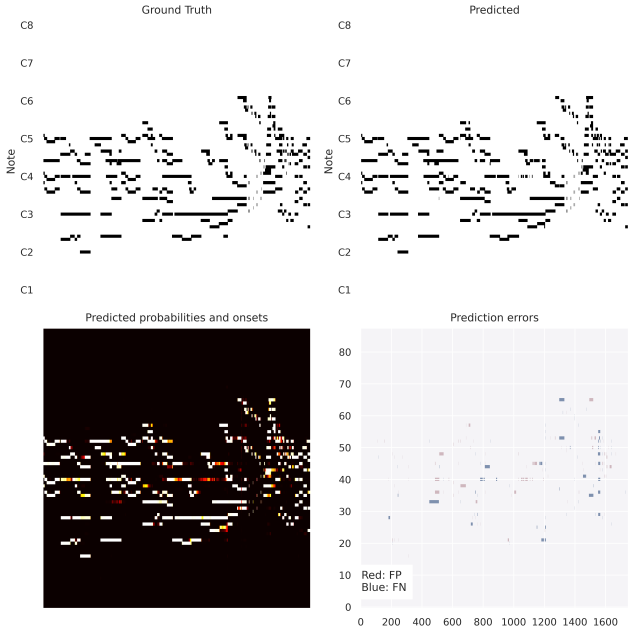


Figure 9: Orpheus model predictions

- GPU: 4x NVIDIA GeForce RTX 2080 Ti (1 maximum per experiment)
- CPU: 2x Intel(R) Xeon(R) Bronze 3204 CPU @ 1.90GHz
- RAM: 96GB (32 maximum per experiment)

C TRANSCRIPTION PREDICTIONS

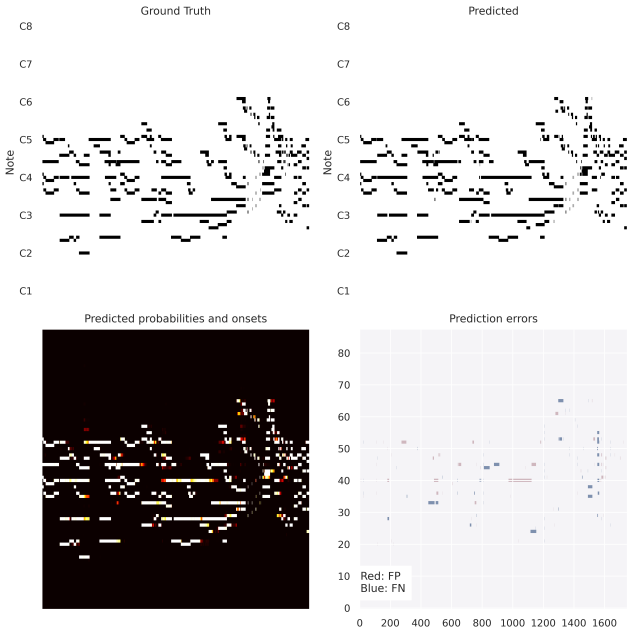


Figure 7: O&F model predictions

Figures 7, 8 and 9 show piano roll visualizations of the transcription performance of our reconstruction of the O&F model (baseline), the baseline with a Transcription Error Correction BERT head on top, and our Orpheus model respectively, on a part from the piece "Fantasy in F-sharp Minor, Op. 28" by composer Felix Mendelssohn.