PROBING THE HIDDEN LAYERS OF A MUSIC GENERATING LANGUAGE MODEL

Rafik Hachana
Innopolis University, Innopolis, 420500, Russia
r.hachana@innopolis.ru

ABSTRACT

Our study delves into FIGARO, a Transformer-based music generation model, revealing its uneven handling of musical elements like chords and velocity. By running a probe experiment on the model's encoder, we uncover a significant bias in processing certain features, directly impacting music generation. An attention visualization further highlights a distinct pattern correlating with musical bars, offering novel insights into music generation with language models.

1 INTRODUCTION

As Transformers (Vaswani et al., 2017) set new performance standards in language modeling, their application to non-NLP tasks becomes more relevant. Since Huang et al. (2018) proposed the Music Transformer, the model became a staple in Music Generation. More recent models like FIGARO (von Rütte et al., 2022) use the same architecture to map a sequence of high-level music features to a symbolic music sequence, which has the potential to offer fine-grained control over music generation. However, Hachana & Khan (2023) showed that not all input feature changes were reflected as expected in the output of FIGARO, which leaves a question mark on the way FIGARO prioritizes the different input features. Hence we run a probe study on FIGARO's encoder hidden state in order to assess the presence of input feature information.

2 RELATED WORK

Most recent music generation studies are based on the Music Transformer (Huang et al., 2018) and generate music in the symbolic MIDI format (Loy, 1985) which is tokenized for the model using representations such as REMI (Huang & Yang, 2020) or CompoundWord (Hsiao et al., 2021). A subset of studies experiment with fine-grained control over music generation (Wu & Yang, 2022; Hadjeres & Crestel, 2020). The FIGARO model (von Rütte et al., 2022) is the state-of-the-art in the fine-grained control task in terms of performance and accessibility as it uses explicit music features for its input.

With a wider adoption of Transformers came the need for interpreting their inner workings. Multiple studies assessed the relevancy of visualizing the Attention weights in the model (Jain & Wallace, 2019; Pruthi et al., 2020; Wiegreffe & Pinter, 2019). This technique is mostly useful in NLP studies (Raganato & Tiedemann, 2018; Vig & Belinkov, 2019). It is more challenging to visualize Attention patterns with more complex input and larger contexts. Probes are a more precise and versatile method to assess the relation between the model's hidden activations and some feature of interest (Alain & Bengio, 2018; Tenney et al., 2019; Hewitt & Manning).

3 METHODOLOGY

We aim to measure the presence of each input feature category in FIGARO's encoder hidden state. We run the experiment on all FIGARO input features except the Time Signature feature (Appendix C).

For each token category, we train a linear probe and a non-linear probe to predict the first ten tokens of the category using the encoder hidden state. The data collection is described in appendix D. We

then compare the test accuracy score over different categories and probe architectures. Training details and hyperparameters are in Appendix B

Additionally, we visualize the Attention activations of a sample from the LakhMIDI dataset to check for patterns that can correspond to the probe results(Appendix A). All source code used in the experiment is publicly available [1]

4 Results and Discussion

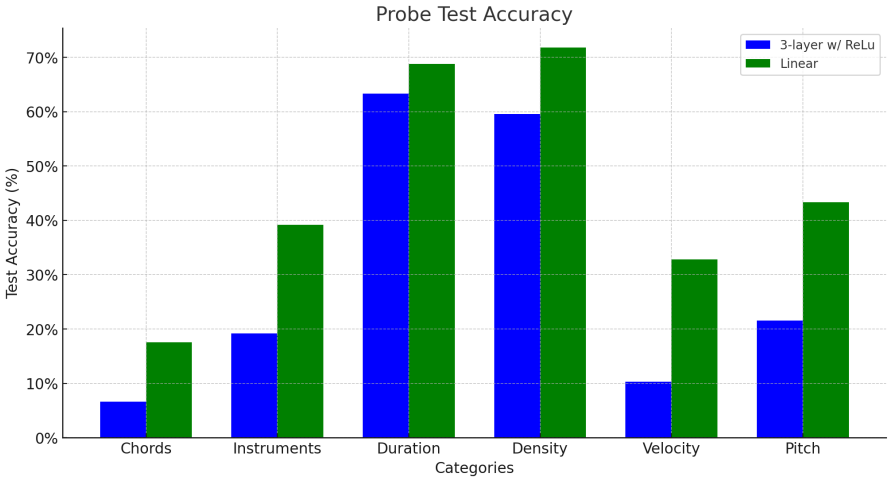

Figure 1: Probe performance across token categories

Figure 1 demonstrates significant performance variations in probe tests across different token categories, with a notable decline in Chords, Instruments, Velocity, and Pitch features. Linear probes exhibit higher accuracy, indicating a strong linear correlation between input and hidden states.

FIGARO's design, based on the independence of token categories representing essential musical features, appears limited in capturing specific attributes such as Chords and Velocity in the hidden state. This limitation aligns with control issues identified in Hachana & Khan (2023). The observed discrepancy in token representation may arise from the input structure or correlations within the training dataset, and can hinder feature control and output diversity.

To address these challenges, solutions could involve adjusting the dataset to reduce feature biases, experimenting with diverse musical feature sets at the input level, and incorporating regularization in training to ensure equal representation of features in the hidden state.

These insights also highlight areas for improvement in language models processing structured languages, which exhibit more rigid patterns than natural languages. While FIGARO successfully identifies bar patterns, as seen in Figure 3, it fails to recognize and represent all token classes within a bar in the hidden state.

5 Conclusion and Future Work

In summary, our analysis of the FIGARO model uncovers significant insights regarding its processing of musical elements. These insights contribute to a better understanding of FIGARO's feature prioritization and inform future improvements in Transformer-based music generation for more sophisticated outputs.

Future research will aim to correlate hidden states with model outputs, extend this probing framework to other language models and tasks, and explore the use of probe loss as a training regularization signal.

[1] `https://anonymous.4open.science/r/figaro-735B/README.md`

ACKNOWLEDGEMENTS

This work was financially supported by The Analytical Center for the Government of the Russian Federation (Agreement No. 70-2021-00143 dd. 01.11.2021, IGK 000000D730324P540002)

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, November 2018. URL http://arxiv.org/abs/1610.01644. arXiv:1610.01644 [cs, stat].

Rafik Hachana and Adil Mehmood Khan. EFFECTS OF SINGLE-ATTRIBUTE CONTROL ON THE MUSIC GENERATED BY FIGARO. 2023.

Gaëtan Hadjeres and Léopold Crestel. Vector Quantized Contrastive Predictive Coding for Template-based Music Generation, April 2020. URL http://arxiv.org/abs/2004.10120. arXiv:2004.10120 [cs, eess].

John Hewitt and Christopher D Manning. A Structural Probe for Finding Syntax in Word Representations.

Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):178–186, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i1.16091. URL https://ojs.aaai.org/index.php/AAAI/article/view/16091. Number: 1.

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. Music Transformer, December 2018. URL http://arxiv.org/abs/1809.04281. arXiv:1809.04281 [cs, eess, stat].

Yu-Siang Huang and Yi-Hsuan Yang. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions, August 2020. URL http://arxiv.org/abs/2002.00212. arXiv:2002.00212 [cs, eess, stat].

Sarthak Jain and Byron C. Wallace. Attention is not Explanation, May 2019. URL http://arxiv.org/abs/1902.10186. arXiv:1902.10186 [cs].

Gareth Loy. Musicians Make a Standard: The MIDI Phenomenon. *Computer Music Journal*, 9(4): 8–26, 1985. ISSN 0148-9267. doi: 10.2307/3679619. URL https://www.jstor.org/stable/3679619. Publisher: The MIT Press.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. Learning to Deceive with Attention-Based Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4782–4793, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.432. URL https://aclanthology.org/2020.acl-main.432.

Alessandro Raganato and Jörg Tiedemann. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL https://aclanthology.org/W18-5431.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT Rediscovers the Classical NLP Pipeline, August 2019. URL http://arxiv.org/abs/1905.05950. arXiv:1905.05950 [cs].

Table 1: Probe training hyperparameters

Hyperparameter	Value
Epochs	100
Batch size	64
Learning rate	3×10^{-3}
Optimizer	Adam
LR scheduler	`ReduceLROnPlateau`
Adam β	(0.9, 0.95)
Grad Norm Clip	1.0

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. pp. 11, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the Structure of Attention in a Transformer Language Model, June 2019. URL http://arxiv.org/abs/1906.04284. arXiv:1906.04284 [cs, stat].

Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. FIGARO: Generating Symbolic Music with Fine-Grained Artistic Control, March 2022. URL http://arxiv.org/abs/2201.10936. arXiv:2201.10936 [cs, eess, stat].

Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation, September 2019. URL http://arxiv.org/abs/1908.04626. arXiv:1908.04626 [cs].

Shih-Lun Wu and Yi-Hsuan Yang. MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer with One Transformer VAE, December 2022. URL http://arxiv.org/abs/2105.04090. arXiv:2105.04090 [cs, eess].

A Attention Visualization

Besides our main experiment, we have visualized the Attention weights for a sample input from the dataset in order to check for patterns that might correspond to the results of the probe experiment. Figure 2 shows the attention weights for the eight heads of the first encoder layer from a sample from the dataset. The weights are displayed on a logarithmic scale.

We have also aligned the heatmap with the types of token from the input description (Figure 3). We observe that the rectangular patterns in the attention heatmap are delimited by the Time Signature token type, which occurs once at the start of each musical bar. Therefore, the rectangular pattern corresponds to the musical bars, which are a fundamental unit in the structure of a musical composition. This means that FIGARO has learned the specific structure of the music from the different token types.

The Attention visualization does not show results that correspond to the probe experiment results. There are no obvious intra-bar patterns. But we can see that different bars of the input descriptions are given different priorities in the encoder.

B Probe training

We train two probes: One linear probe with a single layer, and a non-linear probe with 3 layers and a ReLU activation, and a hidden size of 256. The input size of the probe is always (256×512) (the dimensions of the encoder hidden state). The training hyperparameters are the same for both probes, which are displayed in Table 1. The probes are trained using a CrossEntropy loss.

C FIGARO explained

The FIGARO model (von Rütte et al., 2022) is trained to map a high-level *description* of music to a symbolic representation of music in the MIDI format. This allows it to generate music from a

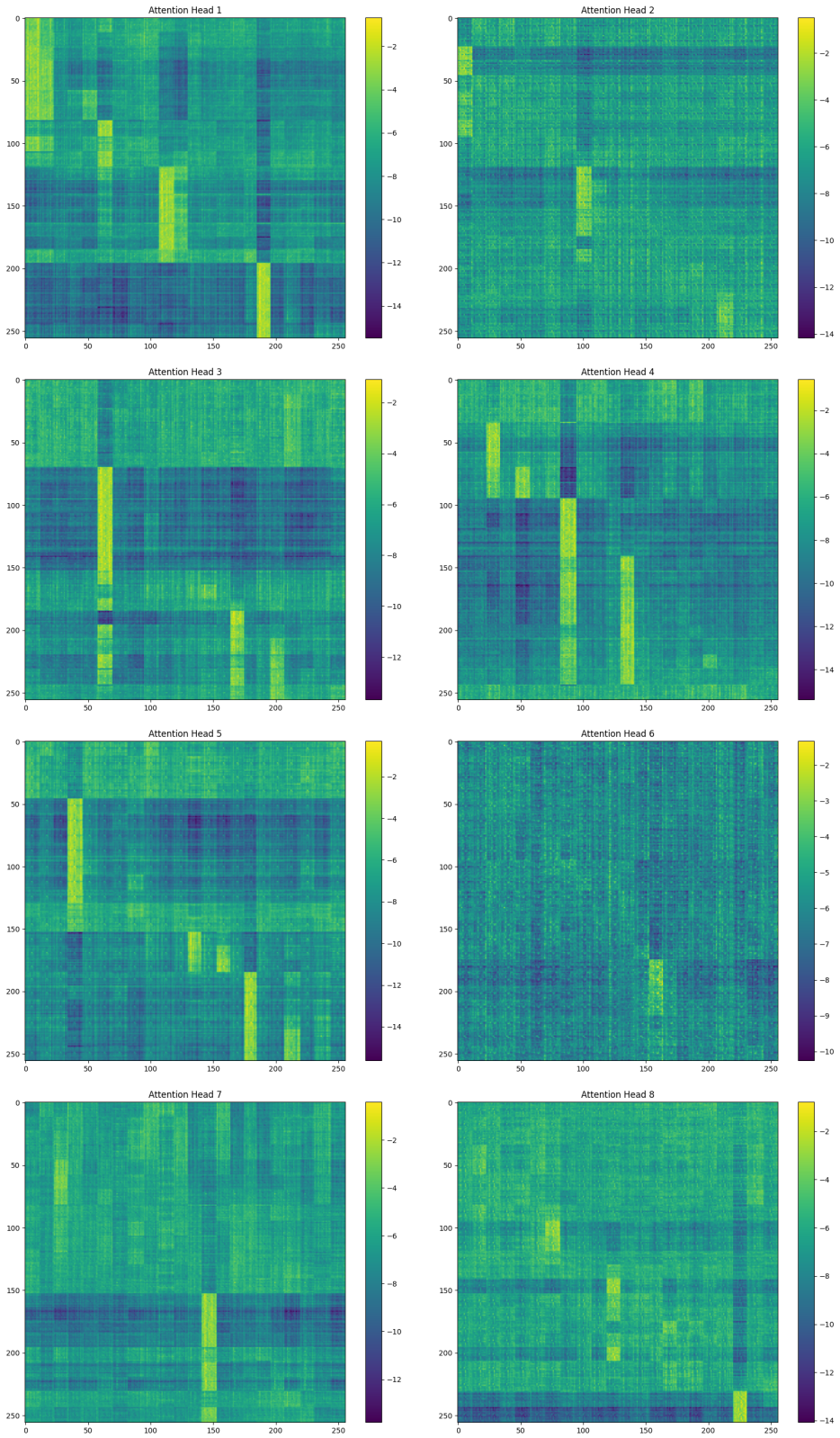

Figure 2: Attention activation heatmaps for the 8 heads of the first encoder layer

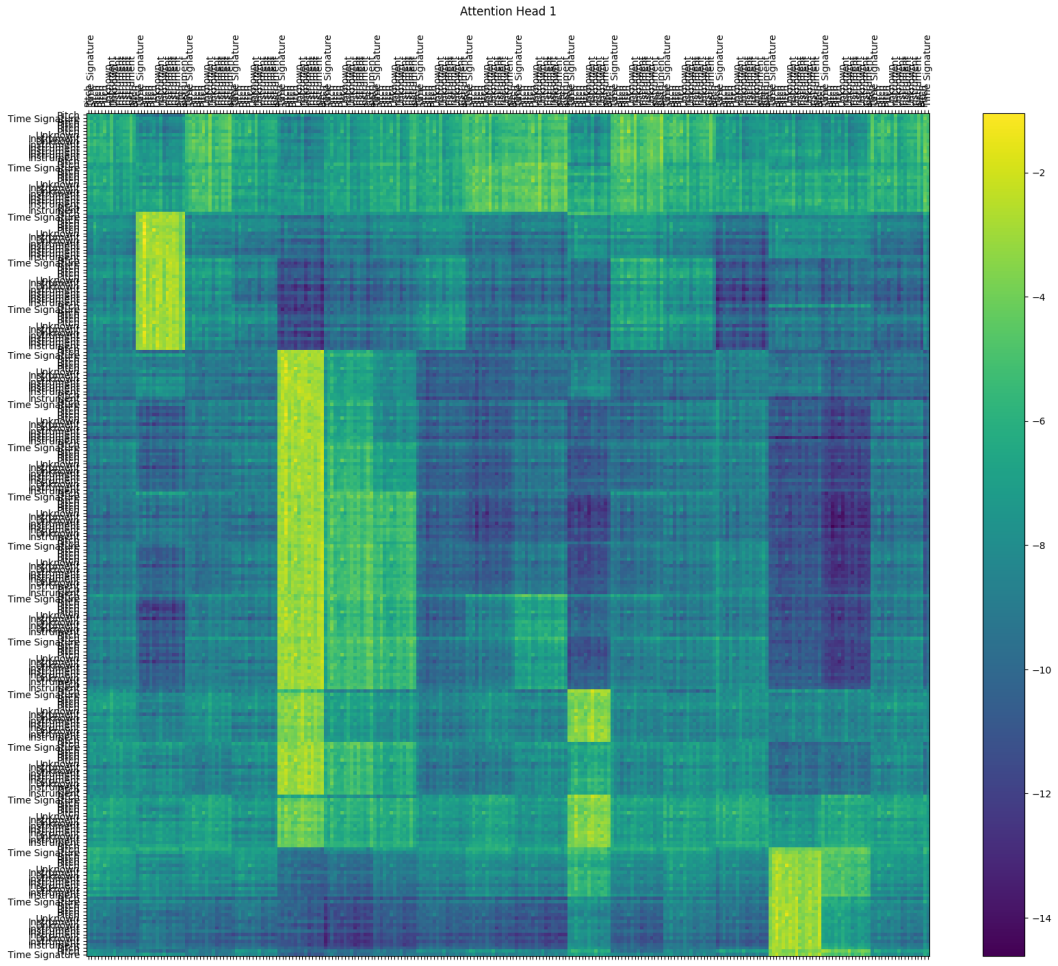

Figure 3: Attention activation heatmap for one head of the first encoder layer, shown with the corresponding token types from the input sequence

list of desired features. The original paper trains multiple versions of FIGARO, with some of them using a sequence of discrete tokens to define the music description, and others using a VQ-VAE during training to generate music from a latent description of MIDI samples. In our experiment, we use the `figaro_expert` checkpoint, which exclusively uses the token sequence description for the model input. In theory, this checkpoint should be able to support fine-grained control of the generated music for all types of description features at any point in the input sequence. The results of Hachana & Khan (2023) show that this does not work as expected for all the input features, which is one of the motivations behind our work. Our results show that the low probe performance for some of the features occurs the most for features that respond poorly to fine-grained control.

The FIGARO input description is a sequence of tokens that follows a strict structure. We include a sample description provided by von Rütte et al. (2022) in Figure 4. The description sequence contains the musical features for different segments of the music. Each described segment has the length of one musical bar. Therefore, the tokens are grouped into bars. Each bar starts with the `bar` token and contains exactly one token for each of the following features: `Mean Pitch`, `Mean Duration`, `Mean Velocity`, `Note Density`, and `Time Signature`. And it can contain multiple tokens for the `Instrument` and `Chord` feature. The token of each feature contains an attached value which can be numerical or categorical. We cover all of the features in our experiment except for `Time Signature`, since this feature is highly biased towards the `4/4` time signature in the dataset.

```
<Bar_1> <Time Signature_4/4> <Note Density_1> <Mean Pitch_13>
<Mean Velocity_19> <Mean Duration_32>
<Instrument_Acoustic Grand Piano>

<Bar_2> <Time Signature_4/4> <Note Density_2> <Mean Pitch_16>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_C:min>

<Bar_3> <Time Signature_4/4> <Note Density_3> <Mean Pitch_16>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_F:dom7>

<Bar_4> <Time Signature_4/4> <Note Density_2> <Mean Pitch_15>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_A#:maj7>

<Bar_5> <Time Signature_4/4> <Note Density_3> <Mean Pitch_15>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_D#:maj>

<Bar_6> <Time Signature_4/4> <Note Density_2> <Mean Pitch_14>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_A:dim>

<Bar_7> <Time Signature_4/4> <Note Density_2> <Mean Pitch_14>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_D:dom7>

<Bar_8> <Time Signature_4/4> <Note Density_2> <Mean Pitch_13>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_G:min7>

<Bar_9> <Time Signature_4/4> <Note Density_3> <Mean Pitch_13>
<Mean Velocity_16> <Mean Duration_32>
<Instrument_Acoustic Grand Piano> <Instrument_Acoustic Bass>
<Instrument_Drums>
<Chord_G:min7>
```

Figure 4: A sample description used to generate music with FIGARO

```
<Mean Pitch_13> <Mean Pitch_16> <Mean Pitch_16> <Mean Pitch_15>
<Mean Pitch_15> <Mean Pitch_14> <Mean Pitch_14> <Mean Pitch_13>
<Mean Pitch_13> <Mean Pitch_13>
```

Figure 5: A sample target from our probe training dataset. This sample is for the probe trained to predict the *Mean Pitch* attribute from the encoder hidden state

D PROBE DATASET

We train a probe for each musical feature to predict a certain musical feature which corresponds to one of the token types of the FIGARO input description (Figure 4). Each sample of the probe training dataset is generated using the following steps:

1. Pick a sample from the LakhMIDI dataset and calculate its description sequence.
2. Run FIGARO on the first 256 tokens of the description and save the encoder hidden state.
3. Extract the first ten occurences of the token type corresponding to the feature of interest from the description and pair them with the encoder hidden state of the sample. An example is shown in Figure 5.

By following the above steps for multiple samples from the LakhMIDI dataset, we obtain a dataset for each feature of interest that pairs the encoder hidden state to the first ten values of the feature.