IMPROVING TOKENIZATION EXPRESSIVENESS WITH PITCH INTERVALS

Mathieu Kermarec[1] **Louis Bigo**[1] **Mikaela Keller**[1,2]
[1] Univ. Lille, CNRS, Centrale Lille
UMR 9189 CRIStAL, F-59000 Lille, France
[2] Inria
{louis.bigo,mikaela.keller}@univ-lille.fr

ABSTRACT

Training sequence models such as transformers with symbolic music requires a representation of music as sequences of atomic elements called tokens. State-of-the-art music tokenizations encode pitch values explicitly, which complicates the ability of a machine learning model to generalize musical knowledge at different keys. We propose tracks for a tokenization encoding pitch intervals rather than pitch values, resulting in transposition invariant representations. The musical expressiveness of this new tokenization is evaluated through two MIR classification tasks: composer classification and end of phrase detection. We release publicly the code produced in this research [1] .

1. INTRODUCTION AND RELATED WORKS

Machine learning, and in particular deep learning, has become a dominant approach to a variety of symbolic MIR tasks including content analysis, classification and generation due to the increasing availability of large corpora and sophisticated neural architectures [1, 2]. A number of Natural Language Processing (NLP) techniques have been twisted for the modelling of music including the self-attention mechanism [3–5], transfer learning [6], or context vectors [7]. This practice is generally justified by the common assimilation of music to a kind of language (*the language of music*) [8, 9], as well as the temporal nature of music which promotes its representation as a sequence of elements, for instance musical notes, that can be processed in a way similar to sequences of words.

NLP challenges have motivated the design of sophisticated neural network architectures, such as LSTM and Transformers, dedicated to the modelling of sequences with long-term relationships between their elements. These elements, called *tokens*, generally correspond to successive words in text. Beyond their use in NLP, these models have successfully shown some ability to model sequences of other type of data including musical content [3]. Representing music as sequences of tokens is however complicated as the organization of elements in the score feature possible simultaneity and overlapping as well as strict timing constraints that are absent from text.

Existing strategies to encode symbolic music into token sequences include the Midi Like tokenization [10] inspired by the syntax of MIDI messages and the REMI tokenization [11] that more closely sticks to the musical score by introducing duration and position tokens. In a more recent tokenization, tokens that define a musical event together are grouped as *compound words* [12]. In order to facilitate the comparison of major tokenizations in MIR research, the MidiTok python library [13] enables the direct translation of any MIDI content into most common tokenizations.

Common tokenizations encode pitch information explicitly, for example with tokens such as `pitch:C3`, which are afterward fed to a sequence model. This contrasts however with a tendency of human listeners to perceive and memorize musical sequences in terms of relative pitches. This limitation arguably complicates the ability of such models to exploit learned musical knowledge across the different keys. This problem is generally tackled by a data-augmentation procedure that consists in transposing the training data in the twelve keys, which dramatically increases the resources required to train the model. As an alternative, we argue for a token representation encoding pitch intervals and therefore invariable to transposition, that would facilitate the uniform transposition of musical knowledge learned by sequence models at any key without any resort to data augmentation.

2. TRANSPOSITION INVARIANT TOKENIZATION

Note features (position, duration and pitch) are grouped in the REMI representation as successive tokens. Starting from this representation, the *uniform pitch-interval* tokenization removes pitch tokens and add pitch interval tokens in between groups of tokens defining a musical note together. Pitch interval tokens represent the semi-tone distance between the two surrounding notes, whether they are played successively or simultaneously. In contrast, the *spatial pitch-interval* tokenization distinguishes pitch in-

[1] https://github.com/MathieuKermarec/improving-tokenization-expressiveness-with-pitch-intervals

Late-Breaking / Demo Session Extended Abstract ISMIR 2022 Conference

	Bach+Liszt	Mozart+Beethoven	Chopin+Schubert	TAVERN
REMI	211	207	210	136
CPW	78k	49k	59k	771

Table 1. Vocabulary sizes of the tokens extracted from several datasets tokenized using REMI or Compound word(CPW). The values for transposition invariant tokenizations are close to those of REMI

tervals between simultaneous and consecutive notes, respectively with horizontal (HPI) and vertical (VPI) pitch interval tokens. In addition, notes occurring at the same onset are systematically ordered in the token sequence by decreasing pitch. Therefore, VPI are exclusively negative values and HPI mostly link high notes which presumably encourages the modelling of melodic features. These two tokenization are illustrated on Figure 1.

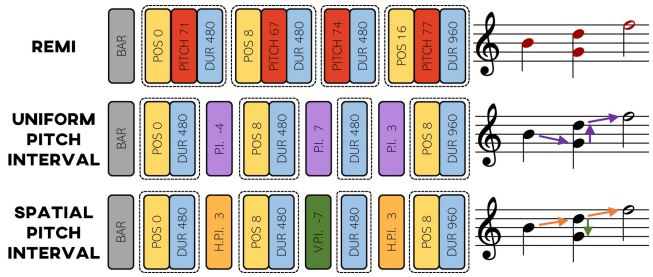

Figure 1. Three tokenizations of a musical sequence. The dotted frames group tokens describing a same note.

3. EVALUATION EXPERIMENTS

3.1 Experimental setting

As an exploratory experiment to study and compare the expressiveness of different tokenizations, we represent each music sequence as a *bag-of-tokens* with TF-IDF weights [2]. Importantly, this representation ignores how the tokens follow one another, resulting in the loss of an essential part of musical information. We argue that this challenging abstraction is a reliable way to highlight the expressiveness of our proposed tokenizations. We deliberately limit our experiments to the use of a logistic regression model in order to facilitate the interpretation of the results and the comparison of the impact of the different tokenizations.

3.2 Classifications Tasks

For the composer classification tasks, we use the GiantMIDI-Piano dataset [14] from which we extracted a total of 740 pieces from 6 composers: Bach, Mozart, Beethoven, Chopin, Schubert and Liszt. Each composer subset is split into a train set and a test set, from which 2000 excerpts of 60 seconds were randomly sampled and tokenized using MidiTok [13] with pitch interval tokenizations plus a baseline *REMI pitch mute* tokenization which is equivalent to REMI but with all pitch tokens replaced by a same token regardless of their pitch value.

[2] term frequency-inverse document frequency : counting the number of occurrences of each token in the sequence and scaling the count by the frequency of the token in the corpus.

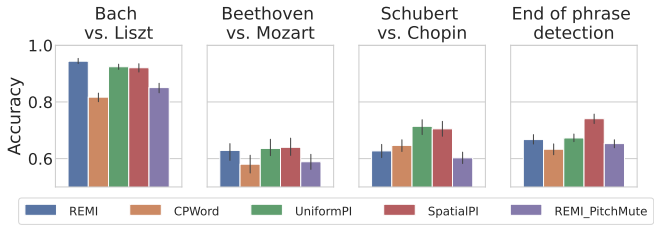

Figure 2. Composer classification and end of phrase detection performed by a logistic regression on TF-IDF token representations, with 5 different tokenizations.

The TAVERN dataset [15] was used for phrase end detection. It includes 27 pieces by Mozart and Beethoven, with phrase boundary annotations resulting in 1060 musical phrases. The dataset is split into a train set and a test set, conserving the original proportion of each composer. The training set was reduced to 1600 chunks of two consecutive bars that were randomly sampled with a balanced representation of chunks including an end of phrase. The test set was similarly reduced into 1100 chunks.

Figure 2 shows that the tasks vary in difficulty and that the choice of tokenization can have dramatic impacts on the performance of the classifiers. In particular, Compound Word tokenization seems poorly suitable for the framework of these experiments, likely because of the high dimensionality induced by its vocabulary of tokens (see Table 1) as compared to the moderate number of training examples. REMI and the Pitch Interval tokenizations have similar performances for composer classification, except for the Schubert vs. Chopin task in which PI tokenizations perform better. We hypothesize that the performance of REMI for the two other classifications is partly due to the pitch range difference between the repertoires of the composers, Liszt and Beethoven arguably employing larger pitch ranges than Bach and Mozart, which is by nature better encoded by absolute pitch tokens. Finally, we see a significative outperformance of the SPI tokenization for the end of phrase detection, presuming a promising ability of this representation to model abstract musical knowledge.

4. CONCLUSIONS AND PERSPECTIVES

We showed that the representation space induced by the choice of a tokenization, for a specific MIR task, can have a strong impact on performance and that transposition invariant tokenizations can improve the musical expressiveness of this space. Future works include extending the comparison of these tokenizations through generative tasks involving the training of a transformer model. We also plan to experiment with other variations of these tokenizations to improve further their musical expressiveness. This includes the encoding of horizontal pitch intervals between the lowest notes of the score, instead of the highest notes, which is expected to improve the modelling of structural features including the detection of end of phrases.

5. ACKNOWLEDGMENTS

The authors are grateful to the Algomus and Magnet teams for fruitful discussions. This work is supported by a special interdisciplinary funding (AIT) from the CRIStAL laboratory and the Merlion PHC Music Language Processing N° 48304SM funded by Campus France.

6. REFERENCES

[1] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep learning techniques for music generation–a survey," *arXiv preprint arXiv:1709.01620*, 2017.

[2] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–30, 2017.

[3] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *International Conference on Learning Representations*, 2018.

[4] M. Keller, G. Loiseau, and L. Bigo, "What musical knowledge does self-attention learn?" in *Proceedings of the 2nd Workshop on NLP for Music and Spoken Audio (NLP4MusA)*, 2021, pp. 6–10.

[5] J. Jiang, G. Xia, and T. Berg-Kirkpatrick, "Discovering music relations with sequential attention," in *Proceedings of the 1st workshop on nlp for music and audio (nlp4musa)*, 2020, pp. 1–5.

[6] Z. Wang and G. Xia, "Musebert: Pre-training of music representation for music understanding and controllable generation," 2021.

[7] C.-H. Chuan, K. Agres, and D. Herremans, "From context to concept: exploring semantic relationships in music with word2vec," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1023–1036, 2020.

[8] D. Cooke, "The language of music," 1959.

[9] R. Jackendoff, "Parallels and nonparallels between language and music," *Music perception*, vol. 26, no. 3, pp. 195–204, 2009.

[10] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, vol. 32, no. 4, pp. 955–967, 2020.

[11] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.

[12] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.

[13] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah-Seghrouchni, and N. Gutowski, "MidiTok: A Python Package for MIDI File Tokenization," in *22nd International Society for Music Information Retrieval Conference*, Online, United States, Nov. 2021. [Online]. Available: https://hal.sorbonne-universite.fr/hal-03418930

[14] Q. Kong, B. Li, J. Chen, and Y. Wang, "Giantmidi-piano: A large-scale MIDI dataset for classical piano music," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 5, no. 1, pp. 87–98, 2022. [Online]. Available: https://doi.org/10.5334/tismir.80

[15] J. Devaney, C. Arthur, N. Condit-Schultz, and K. Nisula, "Theme and variation encodings with roman numerals (tavern): A new data set for symbolic music analysis," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015.