Natural Language Processing Methods for Symbolic Music Generation and Information Retrieval: a Survey

Dinh-Viet-Toan Le, Louis Bigo, Mikaela Keller, Dorien Herremans

Natural Language Processing Methods for Symbolic Music Generation and Information Retrieval: a Survey

DINH-VIET-TOAN LE*, Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
LOUIS BIGO, Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France
MIKAELA KELLER, Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
DORIEN HERREMANS, Singapore University of Technology and Design, Singapore

Abstract – Several adaptations of Transformers models have been developed in various domains since its breakthrough in Natural Language Processing (NLP). This trend has spread into the field of Music Information Retrieval (MIR), including studies processing music data. However, the practice of leveraging NLP tools for symbolic music data is not novel in MIR. Music has been frequently compared to language, as they share several similarities, including sequential representations of text and music. These analogies are also reflected through similar tasks in MIR and NLP.

This survey reviews NLP methods applied to symbolic music generation and information retrieval studies following two axes. We first propose an overview of representations of symbolic music adapted from natural language sequential representations. Such representations are designed by considering the specificities of symbolic music. These representations are then processed by models. Such models, possibly originally developed for text and adapted for symbolic music, are trained on various tasks. We describe these models, in particular deep learning models, through different prisms, highlighting music-specialized mechanisms. We finally present a discussion surrounding the effective use of NLP tools for symbolic music data. This includes technical issues regarding NLP methods and fundamental differences between text and music, which may open several doors for further research into more effectively adapting NLP tools to symbolic MIR.

Keywords: Music Information Retrieval, Natural Language Processing, Symbolic music, Music generation, Music analysis, Deep learning.

Contents

*email: dinhviettoan.le@univ-lille.fr

1 INTRODUCTION

The evolution of Natural Language Processing (NLP) has been marked by a substantial journey, progressing from rudimentary rule-based systems like ELIZA [195] in 1966 to the widespread adoption of sophisticated deep learning models by the general public, such as ChatGPT. In parallel with these advancements, computational music research has been adapting NLP methods for musical data for various analysis and generative tasks. This transfer of NLP methods to symbolic music data has become more and more prevalent in the Music Information Retrieval (MIR) community, especially with the breakthrough of Transformer models.

Natural Language Processing (NLP) is a field at the crossroads between linguistics and computer science that focuses on the interaction between computers and human language. Its main purpose is to understand, interpret, and generate human language taking into account its characteristics, such as syntactic or semantic properties. Through various techniques, in particular, by training deep learning models, multiple tasks are tackled from text analysis such as sentiment analysis, part-of-speech tagging, text similarity or language identification to generative tasks such as summarization, question answering, chatbot conversation, or machine translation.

The field of *Music Information Retrieval* (MIR) combines aspects of musicology and computer science to develop techniques to analyze music, retrieve music-related data, or generate music. While audio files capture music as sound, seen at a low level and described by objects such as waveforms or spectrograms, *symbolic music* depicts music as abstract notations operating on concepts such as notes, chords, intervals, etc. that compose musical scores. Although requiring more sophisticated notation systems, symbolic music representations allow for the study of music at a higher level, such as analysis of harmony, form, or texture. In practice, symbolic music remains prevalent in digital music production mainly relying on the MIDI format, which stands as a ubiquitous standard within digital audio workstations (DAWs). This survey will only consider music viewed as *symbolic* representations.

1.1 Music and natural language: similarities and specificities

Beyond computer science studies, parallels between music and natural language are often drawn, as music is often considered as a linguistic system [87], sharing communication purposes as well as structural similarities with natural language. These parallels are also found in terms of tasks studied in the NLP and MIR fields.

Music seen as a linguistic system · Text representations and symbolic music representations are both semiotic systems [22] based on arrangements of symbols. Text is built on characters and written music can be retranscribed with a variety of symbols derived from various notation systems such as standard notation, numbered notation or tablatures. Both can be represented as sequences of elements which can be segmented or grouped at different levels. Text can be segmented into characters, syntactic phrases and sentences, while music can typically be segmented into temporal units such as notes, motifs, musical phrases, or sections [113] as represented in Figure 1.

Inspired by higher-level concepts in natural language such as grammar or syntax, multiple models of musical syntaxes have been proposed [6, 8]. Such musical grammars rely on intrinsic musical concepts such as tension and relaxation [114] or harmony [166]. These grammatical or syntactic rules lead to expectancy in both language and music [87, 153], inducing similar cognitive reactions for the interlocutor or the listener when they are being transgressed in both language [157] and music [7]. Beyond its formal description, both are specific to human species and are learned through imitation. Both can also be perceived as elements unfolding in time [210] and can be deployed under two modalities: an annotated form (text, sheet music) and an auditory form (speech, musical performance) [53].

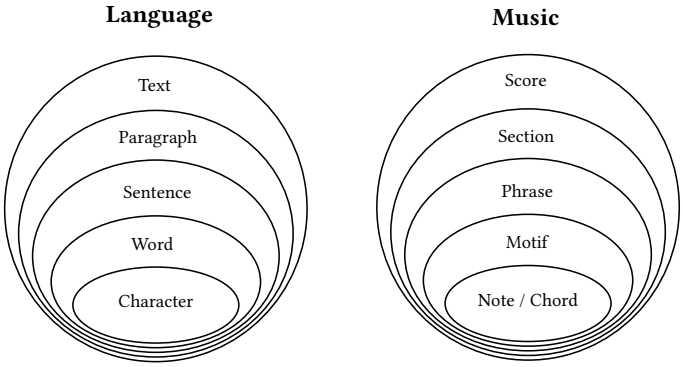

Fig. 1. An *oversimplified* example of segmentation levels in text and symbolic music. Such segmentations can, however, include more or less fine-grained levels and their delimitations can be ambiguous (Section 4).

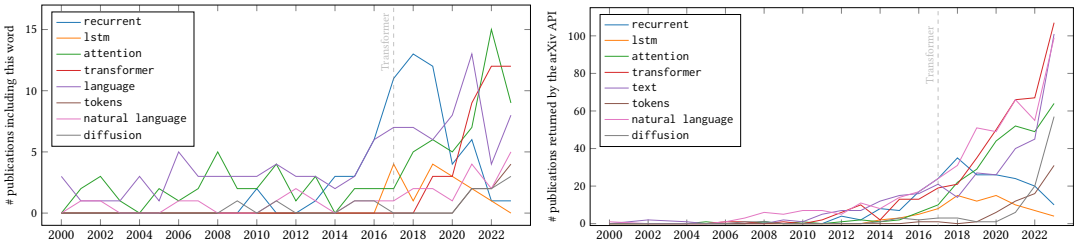

Fig. 2. Evolution of the number of articles containing NLP-related words. (Left) Number of ISMIR papers containing NLP-related words in their abstracts from 2000 to 2023. (Right) Number of arXiv preprints returned by the API query "music AND <term>".

However, major distinctions and particularities still persist between music and text (Section 4), including polyphony (simultaneous musical events), rhythm (rigorous musical time grid) and the multimodal aspect of notes being characterized by multiple musical features (pitch, dynamics, etc.).

Symbolic MIR and NLP tasks · Beyond these parallels between text and symbolic music representations, the Natural Language Processing and Music Information Retrieval research fields are also related by similar tasks they address. On the one hand, commong tasks on *labeled* data in classification of whole textual document or music piece are common tasks, such as music composer classification [156] and text authorship attribution [177], folk song origin classification [75] and language detection [88], music genre [32] and text style [100] classification, or music emotion [86] and sentiment [194] classification. At a lower level, such labels can also describe textual or musical segments which naturally leads to a variety of segmentation tasks in both domains, including musical phrase retrieval [65] or musical form analysis [218] in MIR and discourse parsing [125] or phrase segmentation [84] in NLP.

On the other hand, tasks can rely on *unlabeled* music and text datasets. Apart from clustering tasks in text [206] and music [27], these datasets are usually used to train generative systems following a self-supervised way (*i.e.* predicting parts of the input itself, by learning representations and patterns without external annotations). These models can be trained on tasks such as symbolic music infilling [66] and text infilling [42], or music priming [82] and text continuation [161]. At the scale of a piece or a document, style transfer is performed in both MIR, through musical

genres [203], and NLP, through language high-level elements such as formality or toxicity [94]. More recently, text-conditioned generation has become more and more popular for the general public, including chatbot dialog[1] in NLP, and text-conditioned music generation [132].

These two fields also include numerous tasks that are inherent to one field, as depicted in Figure 3. These tasks specific to each field also reflect the inherent differences between these two types of data, including semantics in text which is key in an entailment task, or polyphony in music which is at the heart of harmonization and accompaniment generation tasks.

1.2 Applying NLP methods in symbolic MIR

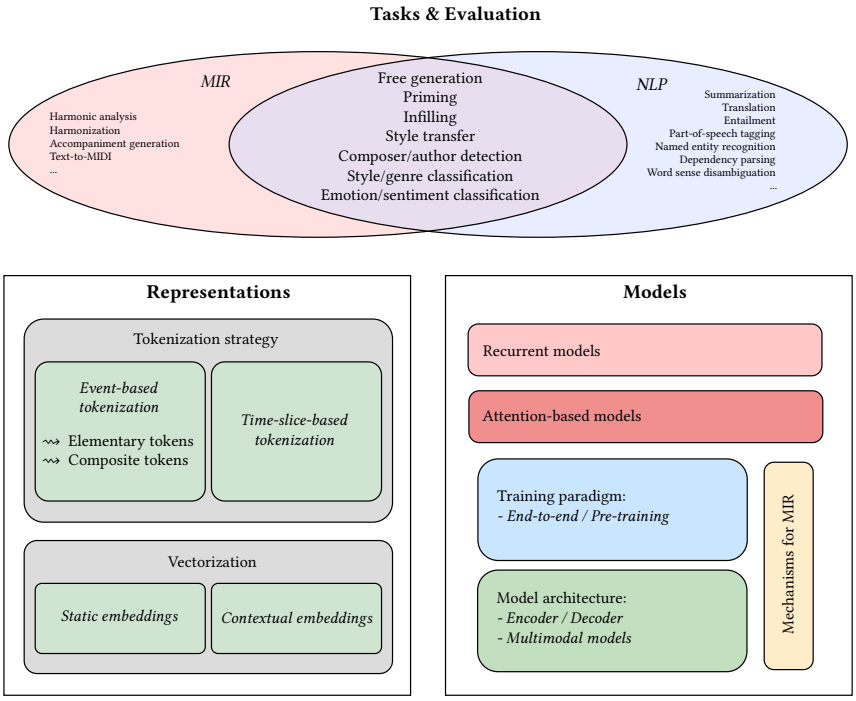

Fig. 3. Overview of the survey, organized around two axes: similarities and specificities of NLP and MIR tasks motivating *representations* of symbolic music inspired by NLP and *models* adapted from NLP for symbolic music.

In the field of symbolic Music Information Retrieval, multiple surveys have been published with a clear focus on music generation. Two main classes of surveys seem to emerge: ones presenting systems through their technical aspects, and ones organizing them based on their musical purpose or task. Firstly, various surveys are driven by the system's technical aspects [51]. More recent surveys now specifically focus on deep learning methods [12]. These surveys organize deep learning generative models following multiple axes such as model architecture [191], types of generation conditions [224], and emotion-driven generation [37]. Instead, several overview articles focus on musical tasks [74] and categorize them based on the nature of the generated content [126] or by the conditions imposed for generation tasks [90].

[1] https://chat.openai.com

An observation drawn from these surveys indicates that the MIR community is closely following new advances in NLP by adapting their tools for music purposes. MIR studies are used to adapting techniques from other fields, such as image processing [80], resulting in this current trend regarding NLP methods for symbolic MIR. Figure 2 describes the number of publications from the ISMIR conference that include NLP-related terms in their abstract as well as music/NLP-related arXiv preprints. In particular, from 2017 with the introduction of Transformers, references to NLP techniques or models have increased drastically so that most of the state-of-the-art models in symbolic music tasks are now based on this model. This phenomenon has encouraged dedicated initiatives in the MIR community, such as the organization of the workshop NLP4MuSA (Workshop on NLP for Music and Spoken Audio)[2] held in 2020 and 2021. In addition, more and more overviews of deep learning approaches for music generation, including NLP-based methods, are presented as tutorials at conferences such as ISMIR[3] or CMMR[4].

The original approach introduced in this survey emphasizes the adaptation of Natural Language Processing methods for music generation and information retrieval within the domain of symbolic music. These encompass tools and methods not only for symbolic music generation but also for existing analysis tasks. From a more epistemological point of view, we hope that analyzing NLP approaches to process symbolic music representations brings an original and promising approach to reconsider the question of what music shares with natural language.

We present an overview of NLP methods adapted for symbolic MIR by proposing taxonomies of two technical aspects (Figure 3): *representations* (Section 2) and *models* (Section 3).

- Choosing a **representation** refers to encoding content (text or symbolic music) into a format suitable for computational processing. Adapting NLP models to symbolic MIR involves mainly sequential representations.
- The **model** performs the task by processing a *representation* of the input content. Such a model can be based on recurrent layers or attention heads, with specific architectures or training paradigms, and potentially implements mechanisms tailored for symbolic music data.

We then discuss the use of such NLP techniques for symbolic MIR by raising possible technical limitations when employing these methods and numerous differences between music and text. We finally outline future directions in which NLP methods can be implemented and adapted for symbolic music (Section 4).

New models or methods adapted from NLP to MIR are released extremely frequently: this survey includes such developments up until the end of 2023. To facilitate continuous updates with these newly released tools, we maintain a collaborative repository accessible at: https://github.com/dinhviettoanle/survey-music-nlp.

2 REPRESENTATIONS OF SYMBOLIC MUSIC AS SEQUENCES

Text data inherently follows a sequential structure composed of elements spanning from individual characters to full sentences. In contrast, representing musical content as a sequence of homogeneous elements is not as straightforward. The multiplicity of information included in a single note (pitch, duration, position, etc.) and the common occurrences of simultaneous notes (polyphony, chords and melody, etc.) make the problem more complex than with text. However, this sequential representation is necessary for the musical data to be subsequently processed by sequential models, which were initially designed to handle text data. This section presents various methods that have been developed to represent *music as sequences of elements*.

[2] https://sites.google.com/view/nlp4musa
[3] http://ismir2023program.ismir.net/tutorials.html#T3
[4] https://cmmr2023.gttm.jp/keynotes/#Yang_abst

2.1 Tokenization strategies

Tokenization refers to the process of representing complex content into a sequence of elements for computational processing. In NLP, tokenization is the task of segmenting a sequence of atomic elements - characters - by grouping them together into informative *tokens* [140], such as subwords, words, or multiple-word expressions. The rise of NLP models in MIR has naturally encouraged the adoption of this term in music representations. We propose a taxonomy of tokenization strategies in symbolic MIR represented in Figure 4.

We organize tokenization strategies within two classes: *time-slice-based tokenization* and *event-based tokenization*. Time plays a special role in music since the time position of notes fundamentally contributes to the conveyed information. Musical elements are intended to occur on an isochronic grid [87] in which notes have rigorous annotated timings on sheet music[5]. Representing time properties of musical elements has led to multiple approaches [12, §4.8] including representations based on regular time steps (Section 2.1.1), or driven by events occurring through time (Section 2.1.2).

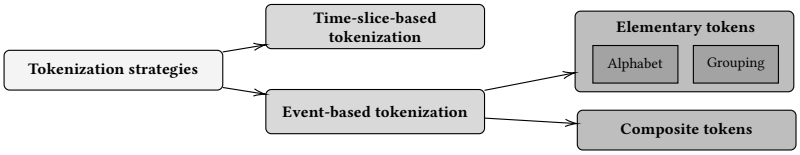

Fig. 4. Taxonomy of tokenizations for symbolic music. Tokens are either based on regular time-slices or events. Among event-based tokenization strategies, tokens encode various features of these events: composite (or multidimensional) tokens encapsulate all these features in a single token, in contrast with elementary tokens where each musical feature is processed one after the other.

2.1.1 *Time-slice-based tokenization.*

Dividing time at evenly spaced timings is a natural approach to representing music since musical elements are notated on scores at specific timings according to particular rhythms. The approaches described in the following section represent symbolic music as a sequence of fixed-time interval tokens.

DeepBach [69] is a model that aims to generate 4-part chorales, for which time is evenly divided at the level of 16th notes. As the number of simultaneous notes is upper-bounded in 4-part chorales, a time step can be represented as a vector containing 4 pitches. In the same way, a concept of "musical words" defined by slices of three beats is proposed [73, 24] to model musical context and semantic relationships in polyphonic music. Beyond pitches, this time-slice representation can be used in the context of drum music [213]. More generally, these representations can be seen as specific cases of piano rolls. This representation relies on matrices in which the horizontal axis represents time, and pitches are encoded along the vertical axis, with additional characteristics such as velocity as a third dimension. Piano rolls are usually portrayed as an alternative to sequential representations by using matrices. However, a piano roll can be converted into a sequential format by considering it as a sequence of piano roll slices - *i.e.* fixed-size multi-hot vectors containing pitches quantized at a specific duration. These serialized piano rolls consider tokens which can represent a small window of slices around a middle piano roll slice [20], or a full musical bar [14].

2.1.2 *Event-based tokenization.*

Unlike time-slice-based tokenization in which tokens are triggered at each time step, *event-based tokenization strategies* involve tokens occurring when a specific event takes place (*e.g.* a note being played, the start of a measure, etc.). Most

[5]Such exact timings can, however, be altered in a performance context where musicians have the freedom to distort this time grid leading to expressive effects such as *rubato*, *accelerando*, or *ritardando*.

tokenization strategies have shifted towards this event-centric approach, helped by the increasing amount of available MIDI data. The MIDI protocol (Musical Instrument Digital Interface) was first developed to handle communication between music software and hardware. The serial transmission of MIDI messages provides a natural way to encode music as sequences of events. The large adoption of this format in the music community has led to the availability of multiple datasets [90] which are essential for training deep learning models.

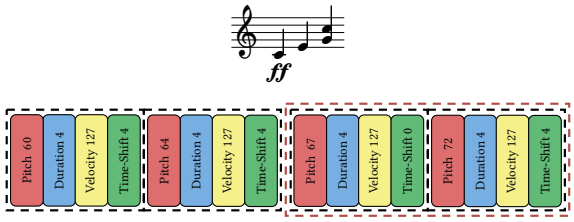

Fig. 5. Artificial sequentiality possibly introduced in a tokenization strategy. By restraining the attributes of a note to pitch, duration, velocity, and time-shift, the sequentiality of the blocks (black dashed blocks) follows the temporality, but the order of the inner musical features is arbitrary. The sequentiality of these blocks can even be artificial for simultaneous events (red dashed block).

However, MIDI messages are of different types: unlike characters in text, musical notes include multiple features, such as duration, pitch, or velocity. Since these features characterize a single temporal event, representing such features sequentially may necessitate introducing an "artificial" sequentiality on top of the temporal sequentiality as illustrated in Figure 5. This sequentiality is even more artificial when representing simultaneous notes occurring at the same time. In the MIR field, two main classes of event-based tokens stand out that we refer to as *elementary tokens* (Table 1) and *composite tokens* (Table 2). Sequences of elementary tokens explicitly integrate this artificial sequentiality where each token is a single musical feature. This can possibly result in two adjacent tokens describing the same temporal event (*e.g.* the pitch of a note followed by its duration). On the contrary, sequences of composite tokens partly bypass this artificial sequentiality by considering tokens as objects aggregating all the musical features describing a temporal event in a unique "super-token".

2.1.2.1 Elementary tokens – Music as sequence of individual features.

The constitutive elements of a sequence composed of musical elementary tokens can be described at two levels (Table 1): the choices of an initial *alphabet* of atomic elements encoding different musical features and a *grouping* of these atomic elements into higher level elements, presumably more expressive.

• *Alphabet* – In text, *tokens* frequently denote words or subwords, which themselves are combinations of smaller elements - characters. In the MIR field, *tokens* rather refer to the *atomic* elements of the sequence that constitute what we refer to as an *alphabet*. This alphabet can be composed of a wide range of entities, such as chord labels, notes, decompositions of a note (*e.g.* pitch, duration, etc.), or structural events such as bars. Thus, choosing an alphabet implies choosing a level at which to describe music and a set of attributes to represent it.

• *Grouping strategy* – Atomic elements can be *grouped* together to form more informative elements. These groupings can be established using fixed-size segmentations, statistically derived groupings, or expert-defined rules. In text, atomic elements (characters) are directly merged together to constitute tokens (words or subwords) leading to a vocabulary of increasing size. Similarly, music atomic elements can be grouped together to enrich the vocabulary with more informative tokens.

Table 1. Overview[6] of event-based tokenization strategies based on *elementary* tokens. The "alphabet" describes the types of atomic elements constituting the alphabet with their type. The "data" corresponds to the type of music considered by the indicated article.

Tokenization	Score-based / Perf.-based	Alphabet *(Atomic elements)*		Grouping	Vocab. size	Data
ABC notation [178]	Score	Text alphabet		Bar patching [201]	N/A	Monophonic
MIDI-like [148]	Performance	<Note-ON> *(MIDI value)* <Time-shift> *(absolute time)*	<Note-OFF> *(MIDI value)* <Velocity> *(integer)*	BPE [108, 212] Unigram [108]	388	Piano
LakhNES [43]	Performance	<Note-ON/OFF-[Trk]> *(MIDI value)*	<Time-shift> *(absolute time)*	–	630	Multi-track
REMI [85]	Score	<Pitch> *(MIDI value)* <Velocity> *(integer)* <Bar>	<Duration> *(music time)* <Chord> *(class)* <Position> *(music time)*	BPE [55, 108, 212] Unigram [108]	332	Piano
REMI+ [187]	Score	REMI alphabet + features: <Time-Signature> *(class)*	<Instrument> *(class)* <Tempo> *(integer)*	–	N/A	Multi-track
Lee et al. [111] *ComMU*	Score	REMI alphabet + metadata: <Instrument> *(class)* <Time-Signature> *(class)* <Min/Max-velocity> *(integer)* <Pitch-range> *(class)*	<Key> *(class)* <BPM> *(integer)* <Rhythm> *(class)* <Number-of-measures> *(number)*	–	728	Multi-track
MusIAC [66]	Score	REMI alphabet + control info: <Tensile-train> *(class)* <Density> *(class)*	<Occupation> *(class)* <Cloud diameter> *(class)* <Polyphony> *(class)*	–	360	Multi-track
Wu and Yang [203] *(MuseMorphose)*	Score	<Pitch-[Trk]> *(MIDI value)* <Velocity-[Trk]> *(integer)* <Bar>	<Duration-[Trk]> *(music time)* <Position> *(music time)* <Tempo> *(integer)*	–	3440	Multi-track
MultiTrack [50]	Performance	<Start-piece> <Start-bar>/<End-bar> <Note-ON/OFF> *(MIDI value)* <Instrument> *(class)*	<Start-track>/<End-track> <Start-fill>/<End-fill> <Time-shift> *(absolute time)* <Density level> *(integer)*	–	440	Multi-track
MMR [127] *(SymphonyNet)*	Score	<Start-score>/<End-score> <Chord> *(class)* <Position> *(integer)* <Duration> *(music time)*	<Start-bar>/<End-bar> <Change-track> <Pitch> *(MIDI value)*	BPE [127]	N/A	Multi-track
TSD [55]	Performance	<Pitch> *(MIDI value)* <Duration> *(absolute time)* <Rest> *(absolute time)*	<Velocity> *(integer)* <Time-shift> *(absolute time)* <Program> *(class)*	BPE [55]	249	Multi-track
Structured [68]	Performance	<Pitch> *(MIDI value)* <Duration> *(absolute time)*	<Velocity> *(integer)* <Time-shift> *(absolute time)*	–	428	Piano
Li et al. [121]	Score	<Pitch-class> *(class)* <Bar> *(integer)* <Duration> *(music time)*	<Octave> *(integer)* <Position> *(music time)* <Velocity> *(integer)*	–	N/A	Monophonic
Chen et al. [18]	Score (Tablatures)	<Pitch> *(MIDI value)* <Bar> *(integer)* <String> *(integer)* <Technique> *(class)* <Velocity> *(integer)*	<Duration> *(music time)* <Position> *(music time)* <Fret> *(integer)* <Grooving> *(class)*	–	231	Guitar
DadaGP [168]	Score (Tablatures)	<start>/<end> <Instrument:note> *(MIDI value)* <String> *(integer)* <Effect> *(class)*	<Wait> *(integer)* <Drums:note> *(MIDI value)* <Fret> *(integer)*	BPE [108] Unigram [108]	2140	Guitar

• ***Building an* alphabet *of atomic elements to encode music*** · A distinction between "MIDI Score" and "MIDI Performance" can be underlined [148]: the first one is a MIDI file converted from a sheet music format (`musicXML`, `kern`...) exactly following a written metrical grid, while the second one is a performance encoded into the MIDI protocol. Scores include information such as exact timings and enharmonics, whereas performance data includes velocity and

[6] An up-to-date and collaborative version of this table can be found at: https://github.com/dinhviettoanle/survey-music-nlp#event-based-tokenization

performance variations such as local tempo or dynamics. In the following, we follow this distinction to organize existing alphabets for symbolic music tokenization.

On the one hand, *performance-based* tokenization focuses on encoding music as sequences of performance events, nearly translating the gesture of an on-stage performer. The MIDI-like tokenization [82] follows MIDI events from the basic MIDI protocol, including a vocabulary of 4 event types: `<note_on>`, `<note_off>`, `<time_shift>`, and `<velocity>`. This tokenization can be adapted for monophonic melodies [165] or a polyphonic ensemble with a fixed number of instruments [43] by having `<note_on/off>` tokens specific to each instrument. TSD (Time-Shift-Duration) [55] adapts the MIDI-like tokenization, using `<duration>` and `<time_shift>` to replace pairs of `<note_on/off>` The Structured MIDI encoding [68] is similar to TSD but enforces the order of tokens describing a same event. This avoids syntax errors in the context of live music generation and improves token sequence consistency by implicitly reducing the vocabulary size at each generation step.

Instead, *score-based* tokenizations describe music as a time-structured system based on multiple discretization levels of time. REMI (Revamped MIDI-derived events) [85] uses a set of score-related elements to tokenize musical data, in particular `<bar>`, `<position>` and `<duration>` both being expressed in musical time instead of absolute timings. The use of such time encoding appears to bring consistency in rhythm. Pitch encodings have also been adapted based on domain knowledge, by relying on pitch classes and octaves instead of raw MIDI numbers. This pitch encoding appears to provide better pitch distributions in both analysis [122] and generative tasks [121]. Multiple extensions of REMI have been implemented, adding additional tokens including metadata [111], musical features [187, 176], control tokens [66], hand positioning for piano music [63] or track information [203]. Before MIDI-based tokenization, early representations of music as sequences rely on score elements [28]. This representation, called "viewpoints", describes relations between successive events, such as melodic contours or positions of events in a bar. The ABC notation has also been used as a direct way of encoding monophonic scores [178] where tokens are considered to be text characters. Basic NLP models can be simply trained on these textual data for generation [178].

In addition, some specificities related to the instrument or the type of music data may prompt the need for adjustments to the tokenization strategy. Tokenization strategies for guitar tablatures have been proposed for generation tasks directly in the tablature space [18, 168] by adding guitar-specific tokens. Moreover, unlike text in language, which consists of a unique stream of words, the challenge of encoding *multi-track* music (*i.e.* multi-instrument, with potentially polyphonic tracks) involves finding a way to represent simultaneous streams as a single sequence of tokens. The representations from MMM (Multi-track Music Machine) [50], MuMIDI [164] and the MMR (Multi-track Multi-instrument Repeatable) representation [127] deal with this issue by adding a token related to tracks. However, MMR and MuMIDI interleave the different tracks to represent the multiple tracks into one sequence. Instead, MMM concatenates all the tracks horizontally to get this single sequence. In other words, comparing these multi-track tokenizations, MMM has a horizontal reading of the score by concatenating single-instrument tracks, while MMR and MuMIDI have a vertical reading of the score by firstly concatenating simultaneous measures or events from multiple tracks.

• **Grouping *atomic elements for shorter sequences and more informative tokens*** · When comparing text and music, textual sentences are often composed of hundreds of characters or around a dozen words, which is an amount of tokens that models such as Transformers can handle well. In contrast, musical sequences may be considerably longer due to various factors such as polyphony or multiple existing token types. To address this complexity issue, two approaches can be considered: adapting the model mechanisms to handle this type of data (Section 3) or manipulating the representation of music in order to compress the sequence length by *grouping* tokens together.

A textual n-gram [96, Chap. 3] is a sequence of n elements (characters, words, etc.) grouped together based on a fixed number of elements to constitute a token. N-grams have been one of the earliest representations of music borrowed from NLP [48], then improved by n-grams/skip-grams [172]. However, while grouping characters is straightforward for text data, musical n-grams can be of a diverse nature with groupings occurring at multiple levels. Musical n-grams can be composed of note intervals or rhythm ratios [198], musical descriptors [28], or chord n-grams to represent music through harmony [147]. These musical n-grams also show statistical phenomena initially observed in text data representations. Various laws such as the Heaps' law [174] or the Zipf's law [198, 155] can be observed with musical n-grams. Musically-informed groupings can be derived from the musical structure of a sequence. The CLaMP model [201], which is based on the ABC notation that includes pipe characters to represent bars, considers a measure-based grouping. Such musically-informed groupings are, however, little studied because note-level groupings are more suited as composite tokens (Section 2.1.2.2), and higher-level structures, such as motifs or phrases, are often not well defined.

Finally, NLP studies have developed *subword tokenization* methods [140] where a vocabulary of subwords is statistically learned on a training corpus. These include Byte-Pair Encoding (BPE) [58, 173], WordPiece [171] or UnigramLM [107]. Some of them have been adapted for music to create musical subwords as tokens. The BPE algorithm is adapted for orchestral data [127] by exploiting the invariance of note order within a chord, to shorten sequence lengths. More than a simple tool for shortening sequences, BPE has also been studied for its specific effects on musical data. Multiple studies applied it on multiple encodings in order to examine how training Transformer models with input reduced by BPE affects both generation and analysis tasks. Although BPE builds a more structured embedding space [55], experiments studying the impact of BPE in music analysis tasks do not show a significant increase in performance [212], unlike BPE applied to text [173]. Finally, UnigramLM subword tokenization is also specifically evaluated on music generation, applied to score-based music and guitar tablatures [108]. Their findings indicate that both approaches contribute to improved data representation, enhance the structural quality of generated music, and enable the generation of longer sequences.

2.1.2.2 Composite tokens – Music as sequence of combinations of multiple musical features.

While sequences of elementary tokens have to introduce an artificial sequentiality by ordering musical features that describe a single event, *composite tokens* encapsulate the entirety of a temporal event by combining all the musical features of this event into a single "super-token". The choice of the nature of the super-tokens, of which musical features to encapsulate into them, and of how the vector representing each super-token is constructed are the main variables in the approaches reviewed in the following. Table 2 describes the type of super-token and the list of features for each approach.

On the one hand, homogeneous super-tokens denote a representation where each super-token contains the same set of features no matter the nature of the event it describes. The representation developed by Zixun et al. [225] is based on the concatenation of multiple one-hot vectors describing pitch, duration, chords, and bar. Octuple [211] is instead based on the embedding of 8 musical features which are concatenated to form the single vector representing a single note. Such homogeneous representations are also used by PiRhDy [122] encoding pitch classes and octave instead of MIDI value, and MMT [44] for multi-track music. Instead of vectors, MuseBERT [192] embeds matrices derived from a set of onset, pitch, and duration aiming at describing both musical attributes with their relations. Beyond notes, the Chordinator model [36] encodes chords described by a root, a nature, extensions, and a set of notes composing the chord.

Table 2. Overview[7] of event-based tokenization strategies based on *composite* tokens. The "musical features" column describes the components of the vectors considered as tokens, in terms of musical attribute. The "embedded object" denotes the manner these musical features are grouped together to form the super-token, including fixed-size vectors or based on event families.

Tokenization	Musical features		Super-token nature	Data
Zhang [215]	<Pitch> *(integer)* <Program> *(class)*	<Velocity> *(integer)*	Homogeneous	Multi-track
PiRhDy [122]	<Chroma> *(class)* <Note-state> *(class)* <Inter-onset-interval> *(music time)*	<Octave> *(integer)* <Velocity> *(integer)*	Homogeneous	Multi-track
Zixun et al. [225]	<Pitch> *(one-hot)* <Current/Next-chord> *(one-hot)*	<Duration> *(one-hot)* <Bar> *(one-hot)*	Homogeneous	Lead sheet
Octuple [211]	<Time-signature> *(class)* <Bar> *(integer)* <Instrument> *(class)* <Duration> *(music time)*	<Tempo> *(integer)* <Position> *(music time)* <Pitch> *(MIDI value)* <Velocity> *(integer)*	Homogeneous	Multi-track
Dong et al. [44] *(MMT)*	<Type> *(class)* <Position> *(music time)* <Duration> *(music time)*	<Beat> *(integer)* <Pitch> *(MIDI value)* <Instrument> *(class)*	Homogeneous	Multi-track
Dalmazzo et al. [36] *(Chordinator)*	<Chord-root> *(class)* <Chord-extensions> *(class)* <MIDI-array> *(multi-hot)*	<Chord-nature> *(class)* <Slash-chord> *(boolean)*	Homogeneous	Chord sequences
Wang and Xia [192] *(MuseBERT)*	<Onset> *(music time)* <Duration> *(music time)*	<Pitch> *(MIDI value)* + factorized properties	Homogeneous	Multi-track
MuMIDI [164]	<Bar> <Tempo> *(integer)* <Chord> *(class)* <Velocity> *(integer)*	<Position> *(music time)* <Track> *(class)* <Pitch / Drum> *(MIDI value)* <Duration> *(music time)*	Family-based	Multi-track
Compound Word [78]	<Family> *(class)* <Bar> *(integer)* <Chord> *(class)* <Pitch> *(MIDI value)* <Velocity> *(integer)*	<Time-signature> *(class)* <Beat> *(music time)* <Tempo> *(integer)* <Duration> *(music time)*	Family-based	Piano
Di et al. [40]	<Type> *(class)* <Strenth> *(class)* <Pitch> *(MIDI value)* <Instrument> *(integer)*	<Beat> *(integer)* <Density> *(class)* <Duration> *(music time)*	Family-based	Multi-track
Makris et al. [135]	Encoder input: <Group> *(class)* <Duration> *(music time or none)* Decoder output: <Onset> *(number)*	<Onset> *(number)* <Type> *(class)* <Value> *(any - depends on type)* <Drums> *(integer)*	Family-based	Encoder: Multi-track Decoder: Drums

On the other hand, methods separating events by families have been developed. This choice is motivated by the fact that a note event is quite different from structural events such as the beginning of a bar. For polyphonic music, MuMIDI [164] represents a token as a sum of the embeddings of bars, position, and tempo, with possibly note characteristics. Similarly, Compound Word [78] gathers tokens into two families: event-related or note-related and concatenates these embedded atomic elements to build the token. It has also been adapted for a task of drum accompaniment generation [135]. This representation is also enhanced by Di et al. [40] in the context of video-to-music, by incorporating a token family related to rhythm, encapsulating rhythm density and strength.

[7] An up-to-date and collaborative version of this table can be found at: https://github.com/dinhviettoanle/survey-music-nlp#composite-tokens

2.2 Comparing tokenization strategies

With all these possibilities of encoding music as sequences, certain tokenization methods may demonstrate better performance on specific tasks or data than others. In NLP, different tokenizers, which initially aim at segmenting text, can result in different vocabularies, so that they can result in unequal performance on various tasks or languages [41]. Few studies have conducted such comparisons between multiple tokenization strategies in MIR contexts. Multiple strategies for pitch (pitch-class vs. absolute) and time grid (time resolution) encodings are compared in the context of monophonic music generation [120]. Fradet et al. [56] focus specifically on time encoding by comparing note positioning and duration encoding on generative, classification, and representation tasks. Beyond tokenization, a comparison between matrix, graph, and sequence representations of symbolic music is performed on various analysis tasks [212].

Technically, the MidiTok Python package [54] has been developed to provide a consistent interface for handling multiple tokenization strategies with various tools designed to manipulate sequential symbolic music data, such as data augmentation or BPE. Multiple other tokenizers derive from this library, including a MusicXML tokenizer [212] or a component integrated into a processing pipeline coupled with the HuggingFace library [108]. Similarly, Musicaiz [72] offers a tokenization framework, with extensive visualization, generation, and analysis frameworks for symbolic music.

2.3 Converting music data into vectors

The previous sections describe music encoded as sequential elements and operations that can be applied to them while keeping their high-level musical meaning. These elements need to be *embedded* or converted into vectors so that the model can process them. Text, subwords, words, or documents need to be projected into a certain space in order to be processed [119] leading to multiple distributional vector space models and embedding methods.

Earliest word representations simply relied on basic one-hot vectors, each with a length equivalent to the vocabulary size. A document is represented by summing all these word vectors, leading to a co-occurrence counts vector, also called bag-of-words (BoW) [96, Chap. 4]. This representation is improved by TF-IDF (Term Frequency–Inverse Document Frequency) [96, Chap. 6] that takes into account the total number of documents in which a word appears. In symbolic music, such BoWs or TF-IDFs have been implemented for music similarity analysis [197], mode classification in Gregorian chant [31], or Chinese folk music clustering [214]. However, these approaches do not capture any sequential information and the resulting space is often sparse, preventing the ability to capture possible proximity between musical elements. Therefore, multiple methods have been developed in the NLP field aiming at projecting words into a dense space including static and contextual embeddings.

Static embeddings assume that each word can be encoded using the same vector regardless of the surrounding context in which the word occurs. Word2Vec [141] is based on a neural network that builds such static embeddings. This method has been adapted for music, implicitly leading to multiple interpretations of the definition of a musical word, including chords or musical phrases. Multiple chord-based Word2Vec have been developed [133, 81]. Such chord embeddings exhibit musical relations and are evaluated on downstream tasks like chord prediction and composer classification [109]. PitchClass2Vec [110] embeds chords with Fasttext [9] which relies on subwords instead of words. In particular, instead of embedding the whole set of pitches constituting a chord, Pitchclass2vec decomposes the chord as intervals in the same way as Fasttext breaks words into n-grams. An alternative approach considers temporal chunks of music as words. Melody2Vec [76] uses Word2Vec on monophonic melodies by assuming such words as musical phrases segmented by GTTM rules [114]. Word2Vec has also been adapted for polyphonic music [73], by considering words as equal-length

and non-overlapping slices of polyphonic music. Visualizing these embeddings shows a structure and organization of the space that follows the rules of tonal harmony [24].

Unlike static embeddings, *contextual embeddings* represent a same word with different vectors depending on the context in which the word occurs because of the polysemous nature of words. Although polysemy and semantics are not directly applicable in music, these contextual embeddings can be useful for symbolic music because the context in which a note appear is fundamental, for instance in functional harmony. Technically, contextual embeddings are built concurrently with model training, such as recurrent or attention-based models (Section 3). Yet, while analyses of learned contextual embeddings are numerous in NLP [128], only very few studies have specifically observed the contextual aspect of such embeddings when applied to symbolic music. Such contextual embeddings have been analyzed from an LSTM model [59] or from BERT embeddings [70]. When comparing GPT-2 and BERT models, the learned embedding space from BERT is shown to be more structured than the GPT's [55]. Musical context can also be defined by the relationship between simultaneous elements, extending beyond the typical temporal context encoded by classic contextual embeddings. PiRhDy embeddings [122] encode such musical-specific context encapsulating melodic and harmonic contexts.

3 NLP MODELS FOR SYMBOLIC MUSIC PROCESSING

An aspect that MIR studies have mainly borrowed from NLP is *models*. This transfer arises primarily from the analogous temporal nature of music, which can be represented as a sequence (Section 2), allowing its processing by NLP-based models. Historically, in NLP, models based on recurrent cells were first implemented in the 1990s, before the breakthrough of attention models in the mid-2010s. MIR studies also followed these trends, adapting these models to symbolic music.

3.1 Recurrent models

Although not based on neural networks, the first sequential models applied to NLP tasks and transferred to music include Hidden Markov Models (HMM) and Conditional Random Fields (CRF) on tasks such as style classification [185] or music generation [183]. In parallel, neural network-based recurrent models (RNN) have been developed and applied for symbolic music [10]. Improvements of basic RNNs have been at the root of several models, such as DeepBach [69] implementing a bidirectional LSTM [77] for chorale harmonization, XiaoIceBand [222] being based on GRU [26] for arrangement generation, or VirtuosoNet [89] implementing a hierarchical RNN [25] with an attention mechanism [3] for expressive performance generation. These recurrent layers are implemented as part of various architectures, from variational auto-encoders with PianoTree-VAE [193] to generative adversarial networks with JazzGAN [182].

However, from the end of the 2010s and the breakthrough of Transformer models [184], the vast majority of state-of-the-art models have now been derived from this model. Although this survey focuses primarily on attention-based models, a thorough overview of recurrent models is available in the online material.

3.2 Attention-based models

Attention is a mechanism proposed by Bahdanau et al. [3], initially as an improvement of RNNs. Vaswani et al. [184] then introduced *Transformers* showing that a model based solely on attention - without using any recurrent mechanism - can outperform state-of-the-art results. More precisely, the model is based on a *self-attention* mechanism and *multi-head attention* blocks. Transformers offer two main improvements to RNNs (Section 3.1). The processing of sequences is sped up, as the entire sequence is passed through the model once and processed in parallel. Moreover, it provides a solution to the problem of vanishing or exploding gradients that occurs with basic RNNs and the issue of hard training with

Table 3. *End-to-end* Transformer-based models applied to symbolic music[8]: such models are directly trained on specific tasks. Models are grouped by architecture. Precisions indicated in the *Representation* column depict the specific adaptations brought to an initial tokenization strategy. The last column indicates if the code is publicly available.

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Decoder-only architecture							
Music Transformer Huang et al. [82]	(2018)	Tf. decoder	Relative attention	Piano / Choral	MIDI-like	Priming Harmonization	✓
Chen et al. [18]	(2020)	Transformer-XL	–	Guitar tabs	REMI-derived (Tablatures)	Free tabs generation	✗
Pop Music Transformer Huang and Yang [85]	(2020)	Transformer-XL	–	Piano	REMI	Priming Free generation	✓
Jazz Transformer Wu and Yang [204]	(2020)	Transformer-XL	–	Lead sheet	REMI-derived (Chords)	Free generation	✓
PopMAG Ren et al. [164]	(2020)	Transformer-XL	–	Multi-track	MuMIDI	Accompaniment generation	✗
Wu et al. [205]	(2020)	Transformer-XL	–	Piano	MIDI-like-derived (composite tokens)	Free generation	✗
Di et al. [40]	(2021)	Tf. decoder	–	Multi-track	CPWord-derived (Rhythm family)	Video-to-music	✓
Chang et al. [17]	(2021)	XLNet	Relative bar encoding	Piano	Compound Word	Infilling	✓
Compound Word Tf. Hsiao et al. [78]	(2021)	Linear Tf. decoder	–	Piano	Compound Word	Priming Free generation	✓
Sarmento et al. [168]	(2021)	Transformer-XL	–	Guitar tabs + multi-track	DadaGP	Metadata-conditioned gen.	✓
Sulun et al. [179]	(2022)	Music Transformer	–	Multi-track	MIDI-like	Emotion-conditioned gen.	✓
ComMU Lee et al. [111]	(2022)	Transformer-XL	–	Multi-track	REMI + metadata	Metadata-conditioned gen. Multi-track combination	✓
SymphonyNet Liu et al. [127]	(2022)	Linear Tf.	3-D positional encoding	Orchestral	MMR	Chord-conditioned generation Priming Free generation	✓
Li et al. [121]	(2023)	Transformer-XL	–	Lead sheet	REMI-derived (pitch class)	Free generation	✗
Multitrack Music Tf. Dong et al. [44]	(2023)	Tf. decoder	–	Orchestral	MMT	Free generation Instr.-conditioned generation Priming	✓
GTR-CTRL Sarmento et al. [169]	(2023)	Transformer-XL	–	Guitar tabs + multi-track	DadaGP	Instr.-conditioned generation Genre-conditioned generation	✗

Table 3. (Continued) *End-to-end* Transformer-based models applied to symbolic music.

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
ShredGP Sarmento et al. [170]	(2023)	Transformer-XL	–	Guitar tabs	DadaGP	Style-conditioned generation	✗
Choir Transformer Zhou et al. [221]	(2023)	Tf. decoder	Relative attention	4-part chorales	Chord + pitch (event-based)	Harmonization	✓
Guo et al. [67]	(2023)	Tf. encoder w/ custom attention	Fundamental music embedding RIPO attention	Monophonic	FME	Priming	✓
Compose & Embellish Wu and Yang [202]	(2023)	Tf. decoder	–	Multi-track	REMI	Lead sheet priming Accompaniment refinement	✓
RHEPP-Transformer Tang et al. [180]	(2023)	Tf. decoder	–	Piano	Octuple	Expressive performance gen.	✓
Angioni et al. [2]	(2023)	Tf. encoder	–	Multi-track	TSD-like	Style classification	✓
Chordinator Dalmazzo et al. [36]	(2023)	minGPT (no pre-training)	–	Chords	Custom chord features (+ MIDI array)	Chord generation	✓
Encoder-only architecture							
MTBert Zhao et al. [218]	(2023)	BERT (no pre-training)	–	4-part chorales	Interval + duration (event-based)	Fugue form analysis	✗
Encoder-decoder architecture							
Transformer-VAE Jiang et al. [93]	(2020)	Tf. encoder-decoder	–	Monophonic	Pitch + duration (time-slice-based)	Priming	✗
Harmony Transformer Chen and Su [19]	(2021)	Tf. encoder-decoder	–	Piano	Piano roll time-slices	Roman Numeral Analysis	✓
Makris et al. [134]	(2021)	Tf. encoder-decoder	–	Lead sheet	Enc.: bar features Dec.: chord + pitch + dur.	Emotion-conditioned gen.	✓
Liutkus et al. [130]	(2021)	Performer	Stochastic positional encoding	Multi-track	REMI MIDI-like-derived (multi-track)	Free generation Groove continuation	✓
Gover and Zewi [63]	(2022)	BART	–	Piano	REMI-derived (hands tokens)	Arrangement generation	✗
Museformer Yu et al. [209]	(2022)	Tf. encoder-decoder w/ custom attention	Fine-/coarse-grained attention Bar selection	Multi-track	REMI	Free generation	✓
Theme Transformer Shih et al. [176]	(2022)	Tf. encoder-decoder	Theme-aligned pos. enc.	Multi-track	REMI-derived (theme tokens)	Theme-conditioned generation	✓
FIGARO von Rütte et al. [187]	(2022)	Tf. encoder-decoder	–	Multi-track	REMI+	Controllable generation	✓

Table 3. (Continued) *End-to-end* Transformer-based models applied to symbolic music.

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
MuseMorphose Wu and Yang [203]	(2023)	Tf. enc + Transformer-XL	In-attention conditioning	Piano	REMI-derived (multi-track)	Style transfer Controllable generation	✓
Accomontage 3 Zhao et al. [219]	(2023)	Tf. encoder-decoder	Instrument embedding	Multi-track	Piano roll time-slices	Accompaniment generation	✓
TeleMelody Ju et al. [95]	(2022)	Tf. encoder-decoder	–	Monophonic	Bar + position + pitch + duration (event-based)	Lyrics-to-melody	✓
MuseCoco Lu et al. [132]	(2023)	Text2Attr.: BERT Attr2Music: Linear Tf.	–	Multi-track	REMI	Text-to-MIDI	✓
Model combinations							
Zhang [215]	(2020)	Generator: Tf. decoder Discriminator: Tf. encoder	–	Multi-track	MIDI-like-derived (composite tokens)	Free generation	✗
Transformer-GAN Muhamed et al. [143]	(2021)	Generator: Tf.-XL Discriminator: BERT	–	Piano	MIDI-like	Free generation	✓
Dai et al. [34]	(2021)	Encoder: Tf. encoder Decoder: LSTM	–	Multi-track	Pitch + rhythm (event-based)	Structure-conditioned gen. Chord conditioned gen.	✗
Choi et al. [21]	(2021)	Chord enc.: Bi-LSTM Rhythm dec.: Tf. decoder Pitch dec.: Tf. decoder	–	Lead sheet	Pitch + rhythm + chord (time-slice-based)	Chord-conditioned generation	✓
Bar Transformer Qin et al. [158]	(2022)	Bi-LSTM - Tf. decoder	–	Lead sheet	Bar + position + melody + chord (time-slice-based)	Free generation	✗
Makris et al. [135]	(2022)	Bi-LSTM - Tf. decoder	–	Multi-track	CPWord-derived	Drums accomp. generation	✓
Neves et al. [145]	(2022)	Generator: Linear Tf. Discriminator: Linear Tf.	Local prediction map	Piano	REMI	Emotion-conditioned gen.	✓
Q&A Zhao et al. [220]	(2023)	PianoTree-VAE Tf. decoder	Instrument embedding	Multi-track	Piano roll time-slices	Accompaniment generation	✓
Duan et al. [49]	(2023)	Generator: Tf. encoder Discriminator: LSTM	–	Monophonic	Pitch + duration + rest (event-based)	Lyrics-to-melody	✗
Video2Music Kang et al. [97]	(2023)	GRU + Tf. encoder-decoder	–	Multi-track	MIDI-like	Video-to-music	✓

[8] An up-to-date and collaborative version of this table can be found at: https://github.com/dinhviettoanle/survey-music-nlp#end-to-end-models

Table 4. *Pre-trained* Transformer-based models applied to symbolic music[9]: such models are pre-trained and then fine-tuned on downstream tasks.

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Encoder-only architecture							
MuseBERT Wang and Xia [192]	(2021)	BERT	Generalized relative pos. enc.	Multi-track	MuseBERT repr.	Controllable generation Chord analysis Accompaniment refinement	✓
MidiBERT-Piano Chou et al. [23]	(2021)	BERT	–	Piano	REMI Compound Word	Melody extraction Velocity prediction Composer classification Emotion classification	✓
MusicBERT Zeng et al. [211]	(2021)	RoBERTa	Bar-level masking	Multi-track*	Octuple	Melody completion Accompaniment suggestion Genre classification Style classification	✓
DBTMPE Qiu et al. [159]	(2021)	Tf. encoder	–	Multi-track	Pitch combinations + duration (event-based)	Style classification	✗
MRBERT Li and Sung [117]	(2023)	BERT	Melody/rhythm cross attention	Lead sheet	Pitch + duration (event-based)	Free generation Infilling Chord analysis	✗
SoloGPBERT Sarmento et al. [170]	(2023)	BERT	–	Guitar tabs	DadaGP	Guitar player classification	✗
Shen et al. [175]	(2023)	MidiBERT-Piano	Pre-training tasks: Quad-attribute masking Key prediction	Multi-track	CPWord simplified	Melody extraction Velocity prediction Composer classification Emotion classification	✗
CLaMP Wu et al. [201]	(2023)	Text enc.: DistilRoBERTa Music enc.: BERT	–	Lead sheet	ABC notation-derived	Text-based semantic music search Music recommandation Music classification	✓
Decoder-only architecture							
LakhNES Donahue et al. [43]	(2019)	Transformer-XL	–	Multi-track	MIDI-like	Free generation	✓
Musenet Payne [152]	(2019)	GPT-2	Timing embedding Structural embedding	Multi-track*	MIDI-like	Priming	✗
MMM Ens and Pasquier [50]	(2020)	GPT-2	–	Multi-track	MultiTrack repr.	Free generation Priming Inpainting Controllable generation	✓
Angioni et al. [2]	(2023)	GPT-2	–	Multi-track	TSD-like	Priming	✓

Table 4. (Continued) *Pre-trained* Transformer-based models applied to symbolic music.

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Zhang et. al [213]	(2023)	GPT-3	–	Drums	Drumroll time-slices	Priming	✓
Bubeck et al. [15]	(2023)	GPT-4	–	Text	ABC notation	Text-to-ABC	✗
Encoder-decoder architecture							
MusIAC Guo et al. [66]	(2022)	Tf. encoder-decoder	–	Multi-track	REMI	Infilling Controllable generation	✓
Li and Sung [118]	(2023)	Tf. encoder-decoder	–	Lead sheet	Pitch + duration (event-based)	Harmony analysis Chord generation	✗
Fu et al. [57]	(2023)	MusicBERT + Music Tf.	–	Multi-track	Octuple	Melody completion Accompaniment suggestion Melody extraction Emotion classification	✗
Multi-MMLG Zhao et al. [217]	(2023)	XLNet + MuseBERT	–	Multi-track	CPWord-derived	Melody extraction	✗
Comparative studies							
Ferreira et al. [52]	(2023)	GRU, Performance-RNN GPT-2 (Tf. decoder) Music Tf. (Tf. decoder) MuseNet (Tf. decoder)	–	Piano	MIDI-like	Free generation	✓
Wu and Sun [200]	(2023)	BERT (Tf. encoder) GPT-2 (Tf. decoder) BART (Tf. enc.-dec.)	–	Lead sheet	ABC notation	Text-to-ABC	✓

Tf.: *Transformer* | Enc.: *Encoder* | Dec.: *Decoder* | Pos. enc.: *Positional Encoding*
(*) These datasets are not publicly available.

[9] An up-to-date and collaborative version of this table can be found at: https://github.com/dinhviettoanle/survey-music-nlp#pre-trained-models

LSTMs. Whereby during back propagation through time, recurrent models tend to struggle in capturing long-term dependencies [146] between words. This phenomenon is also true for music generation [74].

Such models have been applied to symbolic music representations, but also in a variety of other domains, such as computer vision [46] or audio [45]. The use of Transformers has been greatly facilitated with the development of libraries, such as AllenNLP [60], FairSeq [149] or more predominantly, HuggingFace [196]. This library offers model architectures, pre-trained models, tokenizers, and various utilities to simplify the development and deployment of NLP applications. As a result, numerous studies in the field of MIR have started utilizing this library, adapted for musical applications, by leveraging its tools and resources. These include implementations of subword tokenizers, discussed in Section 2.1.2 such as Byte-Pair encoding [173] or Unigram [107] used by Kumar and Sarmento [108] and model implementations such as BERT [39] used in MIDI-BERT [23] or GPT-2 [161] used in MMM [50].

In this section, we propose an overview of these Transformer-based models applied to symbolic music data seen through three technical prisms. A first way of characterizing these models is based on their training paradigm, namely end-to-end training on specific tasks, or pre-training and fine-tuning (Section 3.2.1). In a musical sense, pre-training assumes a hypothesis of a general understanding of music. Beyond the training process, we describe various architectures that have been implemented (Section 3.2.2). The model architecture, based on Transformer encoders, decoders, or combining different types of data, also assumes hypotheses on how music is processed. Finally, we present the enhancements of the Transformers' internal mechanism to specifically process symbolic music data (Section 3.2.3). A summary of these Transformer-based models for symbolic MIR is presented in Tables 3 and 4.

3.2.1 Training paradigms: end-to-end training and pre-training.

Models can first be categorized by their training paradigm. On the one hand, end-to-end models are models trained directly for their specific task. On the other hand, pre-trained models, involve a pre-training of the model for a generic task followed by a fine-tuning step on one or multiple tasks and are at the heart of large language models (LLM) in NLP. From a musical point of view, pre-trained models aim first at modelling or understanding music globally, in the same way as modelling language at a high level in NLP [216], from which specific tasks can then be derived via fine-tuning.

3.2.1.1 End-to-end models.

These models are specifically trained for a particular task, most often, generative tasks. These include Generative Adversarial Networks (GANs) [61] based on Transformers, resulting in models for multi-track generation [215], or emotion-driven generation [145]. Other systems rely on Transformer-based Variational Autoencoders (VAEs) [102] for priming-conditioned generation [93], chord-conditioned generation [21], lyrics-conditioned generation [49] or artistic-controllable generation [187]. This last task is also performed in a multi-track context [111], with fine-grained control of the musical features at the scale of the tracks.

End-to-end models also include several data-specific models designed to process musical data beyond notes. The Chordinator [36] model handles chord data and is based on a minGPT architecture[10], without a pre-training process. Several models are trained on guitar tablatures, for tasks such as tabs generation [18], metadata-conditioned generation [168], style-driven generation [170], or instrument-conditioned generation for bands [169]. Beyond generative tasks, a few models performing analysis tasks have been developed using this end-to-end training fashion. They are trained on labeled datasets, such as roman numeral-annotated datasets [20, 19] or style-annotated datasets [2].

[10] https://github.com/karpathy/minGPT

3.2.1.2 *Pre-trained models.*

In contrast with end-to-end models, pre-trained models are usually not task-specific and follow two training phases. The model is first *pre-trained* on a large corpus of data - generally unlabeled - via generic self-supervised tasks. Once the model is pre-trained, it is *fine-tuned* on a specific downstream task by being trained on a smaller task-specific labeled dataset. This fine-tuning step is also convenient as it requires less data than the pre-training process, and takes less time to train the model instead of multiple trainings from scratch for each existing task. While pre-training was prior to attention-based models, the latest state-of-the-art pre-trained models are now exclusively based on Transformers both in NLP and MIR.

One of the state-of-the-art pre-trained language models is BERT (Bidirectional Encoder Representations from Transformers) [39]. BERT is based on a bidirectional training approach as a masked language model: a pre-training task includes masked word prediction by taking into account its left and right context. Multiple variations of BERT applied to symbolic music have been proposed. MuseBERT [192] develops a specific representation merging musical attributes and relations and processed by the attention mechanism. MusicBERT [211] is a model designed based on RoBERTa [129] and improves the pre-training step by implementing a custom bar-level masking strategy instead of the original token masking. A model combining this MusicBERT model with a Music Transformer has been evaluated on several downstream tasks, resulting in better performances [57]. Instrument-specific BERTs have been implemented such as SoloGPBERT [170] for guitar tablatures, MRBERT [117] for lead sheets or MidiBERT-Piano [23] for piano. This model is then extended beyond piano music and improved with musically meaningful pre-training tasks [175].

GPT (Generative Pre-trained Transformer) [160] is, instead, pre-trained through an auto-regressive task, and is more suitable for tasks involving generation. In NLP, multiple improvements of GPT have been developed such as GPT-2 [161], GPT-3 [13] and GPT-4 [15]. For symbolic music, Musenet [152] and MMM [50] are based on GPT-2 and are trained for conditioned generation. Another approach has been implemented for drum music generation [213]: music is represented as textual data which and a pre-trained textual GPT-3 is fine-tuned on this textual representation of music.

Finally, beyond GPT and BERT, models that integrate pre-trained components have been developed for symbolic music purposes. LakhNES [43] and DBTMPE [159] avoid the lack of data for their respective downstream tasks by being pre-trained on larger corpora and then fine-tuned for chiptune music generation or genre classification.

3.2.2 *Model architecture: Transformer encoder / decoder and multimodal models.*

The model architecture also characterizes the existing attention-based models. In NLP, the architecture proposed by the first Transformer model for translation [184] is based on an encoder-decoder architecture. Afterwards, several NLP models based on either encoders [39], decoders [160], or with modified mechanisms have been proposed. MIR studies have therefore leveraged these existing models to adapt them for symbolic music data. Additionally, unlike NLP models that usually handle text for both input and output, MIR experiments have been conducted with multimodal models capable of processing different types of data, in particular for tasks like text-to-symbolic music. These multimodal models have found application in other domains such as audio processing with MusicLM [1] or image processing with Dall-E [162].

3.2.2.1 *Encoder only.*

Encoders are based on a self-attention mechanism, allowing it to have knowledge of the complete sequence. Bidirectional models, which are based on this encoder-only architecture, have led to symbolic music adaptations of BERT such as MuseBERT [192], MusicBERT [211], MidiBERT-Piano [23], MRBERT [117], and SoloGPBERT [170]. Going further, Han et al. [70] analyze the inner embeddings from BERT when trained on symbolic music and highlight the role of specific

layers on the model performance. BERT is also used as an architecture without its pre-training process by MTBert [218] aiming at analyzing the sections of a fugue form. Beyond BERT, mainly characterized by its pre-training process, Transformer encoders have also been experimented with as a component of global encoder-decoder architecture, in which the encoder keeps a defined role, as detailed below. Such a Transformer encoder is also widely used as the discriminator module in GAN-based models [215, 143, 34], initially developed for generation purposes. Indeed, as most symbolic MIR studies focus on generative tasks, such encoder-only architectures are few in number.

3.2.2.2 *Decoder only.*

In contrast with Transformer encoders, decoders implement a *masked* self-attention mechanism. Such models only have knowledge of past tokens so that they are usually implemented for auto-regressive generative tasks. The first Music Transformer [82] is based on a decoder-only model for priming and harmonization tasks, and is then reused by Sulun et al. [179] for emotion-conditioned generation. Generation is tackled by the MultiTrack Music Transformer [44] for instrument-conditioned generation, the Choir Transformer [221] for 4-part harmonization, Compose & Embellish [202] for lead sheet and accompaniment generation, and by Tang et al. [180] for expressive performance reconstruction. Decoder-only models can also be trained through a pre-training / fine-tuning process, in particular with GPT-based models, such as Musenet [152] or MMM [50]. By comparing multiple decoder-only architectures, such pre-trained decoder-only models appear to perform better in piano generation [52].

Several models combine recurrent models with Transformer decoders. Q&A [220] combines GRU-based PianoTree-VAEs with a Transformer decoder for arrangement generation. In the same way, Choi et al. [21] use a bi-LSTM model as a chord encoder, followed by Transformer decoders as pitch and rhythm generators. This architecture is also implemented in the Bar Transformer model [158] for long-term structure generation, where the LSTM captures note-level dependencies and Transformer decoders capture bar-level relations.

An issue with Transformers is the quadratic complexity of the attention mechanism with respect to the sequence length. The Linear Transformer [98] improves the attention mechanism with a linear complexity. The Compound Word Transformer [78] takes advantage of this computational optimization, coupled with its shorter sequence representation, for piano music generation. SymphonyNet [127] is also based on this model to address the even longer length of orchestral pieces, necessitating this lightweight attention mechanism to effectively process such data. Another improvement of Transformers is Transformer-XL [35], also based on auto-regressive generation, which is able to take into account a much longer context than Transformers. Therefore, such models have been used in several generation studies involving multi-track music [205, 111], piano music [85, 143, 203], lead sheets [204, 121] or guitar tablatures [18, 168, 169, 170]. Chang et al. [17] implement an improved Transformer-XL, XLNet [207], a transformer-based model that can attend to past and future in the same way as BERT, while maintaining an autoregressive predicting order. This model is trained for music infilling.

3.2.2.3 *Encoder-decoder.*

Finally, following the architecture of the vanilla Transformer, multiple models for symbolic MIR implement an encoder-decoder architecture for various tasks. Functional harmony analysis has been tackled by the Harmony Transformer [20, 19]. The model is based on this architecture, in which the encoder has a chord segmentation role while the decoder infers the chord symbol.

For generative purposes, such architectures are used with an encoder which analyzes musical constraints and a decoder that generates musical content. Li and Sung [118] and Makris et al. [134] implement similar architectures, with an encoder analyzing chord (resp. chord valence) that conditions an auto-regressive decoder for a generation task. In the

Theme Transformer model [176], the encoder analyzes the recurrent theme, from which the decoder generates music depending on the conditions regarding the theme position within the generated content. MusIAC [66] is a framework based on an encoder-decoder architecture, in which an encoder is pre-trained as a masked language model, linked with a decoder which performs an infilling task. Multi-MMLG [217] is developed for a melody extraction task. It implements an XLNet model aiming at classifying notes as main melody or accompaniment, followed by a modified MuseBERT model that extracts secondary melodies. In NLP, encoder-decoder models are often implemented for translation purposes [184]. Gover and Zewi [63] implement BART [115], an encoder-decoder architecture with learned positional embeddings, for a task analogous to language translation in the realm of music: music arrangement. This task is also performed by Accomontage-3 [219] for multi-track music with an encoder / multiple decoders architecture. This encoder-decoder architecture is largely used in autoencoder architectures. The Transformer VAE [93] implements a sampling step from a latent space, from which keys and values are derived for the cross-attention mechanism. MuseMorphose [203] and FIGARO [187] are models based on VAEs, developed for controllable generation, which use their latent space representations as constraints.

3.2.2.4 Multimodal models.

Going beyond models handling only a specific type of data, MIR systems have been developed to deal with multiple types of data such as text or video. In symbolic MIR, studies have explored models linking text and music, including a task of lyric-to-melody with TeleMelody [95] processing musical high-level features or Duan et al. [49] operating at the syllable level. Text-to-image systems have been gaining in popularity these last few years resulting naturally in text-to-music systems in both audio [1] and symbolic music. MuseCoco [132] performs this text-to-MIDI task. However, most text-to-symbolic-music tasks currently process an ABC notation, as this encoding is already in a textual format [200]. GPT-4 is able to perform such a text-to-ABC task, among multiple other tasks [15] but struggle at modeling musical concepts such as harmony. Finally, beyond generative tasks, CLaMP [201] integrates two BERT-based models – one for text encoding and the other for music encoding – for an analysis task, namely a tune query task based on natural language descriptions.

Multiple systems have been experimenting with symbolic music generation for video considering the use of music in videos like soundtracks in movies. Di et al. [40] generate music for videos that are analyzed in terms of motion speed and saliency conditioning the generated music rhythm. Kang et al. [97] add a semantic and emotion analysis of the scene, and more specifically generate chords matching these video features.

3.2.3 Adapting attention models inner mechanisms in the context of music.

Extensive studies have been conducted regarding the mechanisms of Transformers applied to text data, including attention and positional encoding. When applied to symbolic music, these mechanisms may be improved to be tailored or visualized for such different data.

Given the human intuitive aspect of visualisation, visualizing different aspects of self-attention (*e.g.* maps, etc.) has been studied. Such visualization can show differences between attention heads being more or less specialized in chords or melody [79]. Self-attention has also been studied as a source of high-level interpretations, such as music theory insights, in terms of motifs, harmony, or temporal dependencies. Such musical objects captured by attention are numerous, including cadential passages [131], musical phrases or modulating sequences [92], or consonant musical intervals [44].

Multiple MIR studies have also developed positional encodings and attention mechanisms customized for the specificities of music. With the Music Transformer model [82], a *relative positional self-attention* mechanism is developed

for music generation enabling the processing of much longer sequences. Similarly, the *stochastic positional encoding* [130] aims to be compatible with linear complexity attention. The specificities of multi-track music inspired the SymphonyNet model to develop a *3-D positional embedding* [127] in which the track order is permutation invariant, unlike note or measure that must remain time-dependant. Musically meaningful positional encodings have been developed based on notes attributes and relations [192], measures [17], musical themes [176], structure and musical time [152], or instruments [220, 219].

The attention mechanism has also been adapted for symbolic music. The Museformer model [209] is based on a *fine-grained and coarse-grained attention* aiming at reducing the complexity of the mechanism, leveraging the expected repetitive aspect of music. The *RIPO (Relative Index, Pitch and Onset) attention* [67] is proposed with the *fundamental music embedding*, relying on the structure of symbolic music built on relative onsets and pitches. In a context of controllable style transfer, the MuseMorphose model [203] includes an *in-attention conditioning* that takes into account constraints in the self-attention computation. For lead sheet data, a melody/rhythm cross attention is implemented in MRBERT [117], in which these two features are merged and simultaneously processed through attention.

Training strategies with musical specificities have also been developed. Based on a GAN architecture [61], a *local prediction map* [145] is proposed so that the discriminator also specifies which parts of the generated sequence is real or generated. Pre-trained models, in particular masked language models, are usually pre-trained on a token prediction task from a masked sequence and a next sentence prediction task [39]. For symbolic music, MusicBERT [211] is pre-trained with a *bar-level masking*: instead of masking a single token and leveraging its Octuple representation, the pre-training process masks a type of feature for all the tokens within a bar. This masking is improved with *quad-attribute masking* [175]. These strategies avoid information leakage between tokens, as some musical features can be easily inferred from adjacent tokens. Taking inspiration from the multi-task pre-training approach of the original BERT model, Shen et al. [175] also propose an analogous pre-training task with next sentence prediction with *key prediction*.

4 DISCUSSIONS AND FUTURE DIRECTIONS

The previous sections outline various NLP approaches adapted to music data, resulting in the development of state-of-the-art tools for multiple symbolic MIR tasks. While these results are shown to be empirically effective, it is worth taking a step back on this practice by questioning the musical appropriation of tools that have originally been thought for natural language. Such issues can either stem from technical challenges, as NLP methods have been specifically developed and tailored for text data, or from high-level considerations, such as inherent differences between text and symbolic music.

4.1 Technical limitations of using NLP methods for symbolic MIR

NLP tools have been developed to specifically process text data, a type of data that remains significantly different from symbolic music, as discussed in Section 4.2. These methods tailored for text data may lead to technical specificities inherent to the field of NLP, which can therefore be questioned when applied to symbolic MIR.

Data availability · Text data differ from symbolic music data by a much wider availability. For example, large language models such as GPT-3 [13] are trained on datasets containing 300 billion tokens. Compared to symbolic music, multiple models [50, 187] are trained on the LakhMIDI dataset which is composed of 175k songs, resulting in only 26M tokens using a basic MIDI-like tokenization. Beyond the quantitative side of symbolic music datasets, there is an unavoidable bias in terms of music style diversity, as classical and pop music is much more numerous than other styles. Moreover, while new text data are released in large amounts, contributing to extending datasets such as CommonCrawl

based on publicly available text, symbolic music data is less likely to be released at this rate. Thus, there is a huge gap between the amount of data needed to train text models, on which Transformers are inherently efficient with such a large amount of data, and the availability of symbolic music data.

Latin alphabet and musical alphabet · The Latin alphabet, on which most NLP studies are based, is composed of homogeneous elements or characters. In contrast, musical alphabets based on the MIDI protocol are heterogeneous, consisting of multiple types of tokens, such as velocity or duration. Therefore, musical notes are based on combinations of these atomic elements. This combinatorial aspect is fundamental in music as two slightly different combinations can lead to radically different notes. In substance, this is comparable to Chinese characters that can be based on different radicals, leading to entirely different meanings [199]. Such models have been developed for Chinese NLP, and take these radicals into account [181].

4.2 Discussing parallels and contrasts between natural language and music

The use of NLP methods in MIR implies that music is associated with a kind of language, which is widely debated in the musicological community. While sharing similarities, several differences distinguish symbolic music from text, including low-level structural properties and organization, and high-level differences, especially regarding their respective function.

4.2.1 Structural differences between text and symbolic music.

The adaptation of NLP tools for MIR is facilitated by several similarities between music representations and text including their sequential organization. Yet, these analogies are limited, as some aspects such as polyphony or rhythm remain inherent to music.

Time dimension in language and music · While speech might have a temporal dimension in terms of speech rate [188], text does not explicitly encode any of these rhythmic modulations. In contrast, musical rhythm is based on an isochronic grid [87] in which notes are notated with rigorous timings, in terms of onsets and durations, beyond some microtimings linked to performance embellishments or tempo changes.

Simultaneity in music · In music, while sequence of notes in monophonic music can be compared to words in text, polyphony adds a dimension that does not find any analogous element in text [7]. Modeling simultaneous events in a one-dimensional sequence requires approximations. Polyphonic music can be considered in two different ways: music can be read vertically by modeling it as a sequence of temporal events which interleaves different parts, or instead, music can be read horizontally by concatenating each part one after the other [112].

Multimodality of music · Musical constitutive elements are less homogeneous than text data. Textual constitutive elements are of a single type: characters and possibly punctuation. In contrast, music symbols combine structural elements (bars, position, etc.), note-related information (pitch, duration, dynamics, etc.) and global information (tempo, instrument, etc.). Regarding computational implementations, this possibly introduces an artificial sequentiality when modeling music because multiple musical features describing one temporal event must be ordered.

Segmenting text and music · While whitepsaces facilitate token segmentation of text in many languages, identifying boundaries of musical motives and phrases remain subjective [124] or can even overlap [71]. In this sense, music might be more easily compared to unsegmented languages [151] where word segmentation can be unclear [83]. Therefore, the application of NLP models that perform well on space-delimited languages in the context of symbolic music can be questioned.

Musical grammar and natural language grammar · While grammar is central for natural language, the existence of a global grammar describing music is also not unanimously accepted, even in a specific style [38]. Multiple grammars have been proposed to describe music from a general point of view, such as GTTM [114] or the implication-realization model [144]. Harmonic concepts have also been modeled as a grammar for music. Such harmonic rules are established by a specific musical style or era [103]: however, something which is considered "regular" in a style can appear as an "irregularity" in another style, while still being considered as music. This absence of "rightness" in music consolidates the idea that aesthetics plays the most prominent role in music [104]. Consequently, in MIR, the evaluation systems performing generative or even analysis tasks can be delicate due to this aesthetic dimension.

4.2.2 Functions of natural language and music.

The question of defining the function of music has been extensively studied and discussed [87, 53, 210]. Communication is central in language because it conveys ideas, thoughts, concepts or propositions. Yet, in music, communication is often considered only one of several functions [138]. This musical communication is often seen as serving other purposes than conveying ideas: the concept of semantics, which is pivotal in language, is missing or at least not essential to the appreciation of music. Music may not carry any literal meaning, or at least that cannot be compared to linguistic meaning [114]. Bernstein declared on this topic [5, p. 33]:

> *Music, of all the arts, stands in a special region, unlit by any star but its own, and utterly without meaning [...] except its own, a meaning in musical terms, not in terms of words.*

Instead, music is more associated with *affect* and serves as an *emotional expression based on aesthetics* [87]. Beyond being provoked by music, this emotional characteristic is sometimes considered as *intrinsic* to the music: some compositional process can represent or symbolize an emotion [16]. While highly influenced by culture, Cooke illustrates this phenomenon by describing third intervals in Western music [30, p. 57]:

> *Western composers, expressing the 'rightness' or happiness by means of the major third, expressed the 'wrongness' of grief by means of the minor third [...].*

This point of view that interprets musical meaning in terms of emotional descriptions is highly debatable, as these considerations often originate from cultural effects [150] or musical education. Indeed, in both language and music, textual signs also do not inherently carry meaning. Instead, meaning is attributed to these signs because a particular community, from a specific era or culture, collectively establishes an agreement to associate a certain set of signs with a particular concept [136, p. 21]. In music, such processes are at the root of *program music* [105]. The question of attaching meaning or semantics to music has been a subject of extensive debate for centuries and is unlikely to have a universal answer. Returning to a technical standpoint, this epistemological debate underscores the need of carefulness when applying NLP tools for symbolic music.

4.3 Future directions

NLP studies have been developed along several axes, including various aspects that may serve as research directions for symbolic MIR studies: lighter models, explainability of representations and models, and task benchmarks.

4.3.1 Towards lighter models.

In the field of NLP, various studies have focused on developing computationally efficient yet lighter models [223], especially with the rise of large language models. Such optimizations leading to lighter models are desired for multiple reasons, including reducing training or inference time, as well as energy consumption or hardware costs. Multiple studies

have explored model compression with knowledge distillation [62]. This distillation process implements a lightweight student network which is trained to reproduce a pre-trained teacher network. In NLP, this has led to lightweight models such as DistilBERT [167]. In contrast with distillation, pruning methods are based on altering an initial model by removing weights. Transformers are shown to be possibly pruned by removing most of the attention heads while keeping decent performance [139] and can help model explainability [186]. Finally, model design optimizations for lightweight processings have been developed such as token skipping in PoWER-BERT [64] or sliding window attention with cache in Mistral 7B [91]. In MIR, such advances towards lighter models have begun to be tackled in the context of audio music [47].

In the field of symbolic MIR, models are currently not as big as NLP models which can reach 175B parameters in the case of GPT-3 [13]. Nevertheless, there is a growing recognition of the efficacy of lighter models for symbolic music data, including the development of Compound Words [78] for smaller sequences, or smaller vocabulary resulting in smaller embeddings [120]. These studies emphasize a promising direction for the application of lighter models in symbolic MIR research. This direction may involve developing light methods specifically tailored for symbolic music, featuring fewer parameters, reduced memory usage, or shorter training or inference times. Such light models can have practical applications in real-time symbolic music generation, including improvisation where an instantaneous inference time is required.

4.3.2 Towards more explainability.

Deep learning models are often perceived as black boxes, lacking explanations for the decisions they make. Several studies address the explainability aspects of NLP tools [216]. From a technical standpoint, retrieving explanations from these tools can take various forms. Extrinsic evaluation of a model involves assessing its performance on probing tasks. In NLP, these probing tasks can vary in nature [29], encompassing syntactic or semantic information retrieval [101]. In contrast, intrinsic evaluation refers to directly analyzing the inner representations occurring in the model. In NLP, intrinsic evaluation is frequently conducted on word embeddings to assess how well a model represents words in relation to each other by examining relations like word similarity or analogies [190]. In the context of Transformers, beyond embeddings, multiple representations can be analyzed [11], in particular attention, being a particularly human-interpretable mechanism.

At a low level, while text representations are most of the time based on words, music representations can be of very different nature. Therefore, specific representations can gain in expressiveness by incorporating more or less musical information [137, 99]. More recently, rationalization (*i.e.* providing a natural language explanation of the process) based on LLMs has been explored to provide musical descriptions of symbolic music data [106]. Going further, providing interpretable tools that align with human behaviour can encounter challenges due to the inherent subjectivity of music. In the context of music composition, stylistic aspects may offer different explanations, and certain passages may only be explained by artistic effects desired by the composer [33]. Despite this subjectivity and artistic aspect present in music, studying the explainability of tools for symbolic music can be a way to gain a better understanding of how models process music data. For instance, analyzing models on simple tasks such as style classification can highlight or confirm musicological characteristics in a particular style. Similarly, with the increasing popularity of text-to-music systems, interpreting models on such tasks may reveal relations between specific words with the resulting generated content, potentially leading to questions regarding biases within the currently available datasets of symbolic music.

4.3.3 *A need for benchmarking and comparative analysis.*

Benchmarks (*i.e.* commonly accepted combinations of datasets, tasks, and evaluation metrics against which new models can be tested) are crucial for meaningful model comparisons. The NLP community has introduced several benchmarks such as GLUE [189] to evaluate language understanding. Other specific NLP benchmarks have also been developed, such as cross-lingual benchmarks [123] or domain-specific benchmarks [154].

In symbolic MIR, there is currently an apparent lack of standardized benchmarks. Though, some symbolic music datasets are recurrently used as training datasets [90], but they rarely come with a set of evaluation tasks. Such standardized bundling of datasets, tasks, and evaluation metrics for symbolic music data, similar to the past MIREX challenges[11], may provide better frameworks to compare and evaluate models. This question of model evaluation is fundamental. Subjectivity is often present in music, both in analysis tasks, such as functional harmony analysis, in which annotator biases can emerge, and in generation tasks. Evaluation of generative systems through listening tests is even more subjective [208], but for which evaluation metrics have been proposed [204, 108]. Valuable contributions regarding these benchmarking issues can be an evaluation toolkit library aiming at retrieving features from generated pieces and comparing them to those extracted from a test set. However, this may explain the challenges in establishing such music benchmarks: the inherent subjectivity of music aesthetics restricts the possibility of "reference data", which are essential for model evaluation.

4.3.4 *Exploring further models for symbolic MIR.*

Beyond improving existing MIR models, several NLP models implement mechanisms or optimizations that can be relevant to symbolic music data. The Longformer model [4] aims to represent long documents by implementing linear complexity attention. Moreover, it also manages to perform well on character-level language modelling tasks. These two characteristics are fundamental in symbolic music, as musical sequences are often longer than textual sequences. Additionally, unlike text where words are often considered as basic tokens, such grouping is less direct in music, so that symbolic music tasks are more similar to textual character-level tasks. On the representation side, BERT-sentence [163] may be relevant in the field of symbolic MIR. This model builds embeddings for entire sentences and performs comparisons between pairs of sentences with a faster computing time. In symbolic music, where a recurrent question concerns music segmentation, such textual sentence-derived representation holds potential relevance. In more practical cases, pattern matching is often used in incipit search engines such as RISM[12]: an embedding-based query method can improve the tool's flexibility.

Finally, beyond NLP and the excitement in the general public for tools based on natural language generation, another trend stemming from research studies is image generation, in particular, text-to-image systems which are based on *diffusion models*. Numerous recent models now integrate state-of-the-art techniques from both domains, using diffusion models coupled with Transformer blocks for controllable generation [116, 142]. Therefore, asobserved in recent publications and preprints (Figure 2), a new trend from recent MIR studies is to adapt models initially developed for images to process music, in the same way as state-of-the-art NLP models have been adapted for symbolic music.

5 CONCLUSION

Symbolic music is frequently associated with natural language, drawing parallels based on structural similarities, especially in their sequential representations and numerous shared tasks. Consequently, the domain of Music Information

[11] https://www.music-ir.org/mirex
[12] https://opac.rism.info

Retrieval, with a specific emphasis on studies centered on symbolic music data, frequently draws inspiration from methods employed in Natural Language Processing. This survey organizes these NLP tools adapted for symbolic music based on two aspects: representations and models.

The process of representing text and symbolic music through sequences, referred to as tokenization, has been widely studied in the MIR field, leading to the development of various tokenization strategies. In contrast with text where words are often considered as basic tokens, the diversity of symbolic music tokenization strategies mainly stems from the multimodality of music, wherein each note can be described by various features. This results in tokenizations based on time slices or musical events, incorporating technical improvements such as token grouping or composite tokens. These representations of symbolic music are then processed by models that draw inspiration from models initially developed to process text. Such models have been historically based on recurrent models until the breakthrough of Transformers in the field of NLP which then spread the development of several attention-based models in the field of symbolic MIR. Nevertheless, acknowledging the particular characteristics of music in comparison with text, numerous models have incorporated music-specific mechanisms into Transformers, such as positional encoding or specialized attention mechanisms.

Despite the great performances of these models on downstream tasks such as generation or information retrieval, this usage of NLP tools - initially tailored for text data - on symbolic music can be questioned. This includes technical issues, but also inherent epistemological differences between text and music. These questions can therefore lead to future directions regarding this current trend, by keeping on taking inspiration from NLP advances, such as lighter, explainable models or benchmarks, to improve tools for symbolic music generation and information retrieval.

REFERENCES

[1] Agostinelli et al. 2023. MusicLM: Generating Music From Text. arXiv:2301.11325.

[2] Angioni, Lincoln-DeCusatis, Ibba, and Recupero. 2023. A Transformers-based Approach for Fine and Coarse-grained Classification and Generation of MIDI Songs and Soundtracks. *PeerJ Computer Science*, 9, e1410.

[3] Bahdanau, Cho, and Bengio. 2015. Neural Machine Translation By Jointly Learning To Align and Translate. In *International Conference on Learning Representations (ICLR)*.

[4] Beltagy, Peters, and Cohan. 2020. Longformer: The Long-Document Transformer. arXiv:2004.05150.

[5] Bernstein. 1959. The Joy of Music. Simon and Schuster.

[6] Bernstein. 1976. The Unanswered Question: Six Talks At Harvard. Vol. 33. Harvard University Press.

[7] Besson and Schön. 2001. Comparison Between Language and Music. *Annals of the New York Academy of Sciences*, 930, 1, 232–258.

[8] Bod. 2002. A Unified Model of Structural Organization in Language and Music. *Journal of Artificial intelligence research*, 17, 289–308.

[9] Bojanowski, Grave, Joulin, and Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.

[10] Boulanger-Lewandowski, Bengio, and Vincent. 2012. Modeling Temporal Dependencies in High-Dimensional Sequences: Application To Polyphonic Music Generation and Transcription. In *International Conference on Machine Learning (ICML)*.

[11] Braşoveanu and Andonie. 2020. Visualizing Transformers for NLP: A Brief Survey. In *2020 24th International Conference Information Visualisation (IV)*, 270–279.

[12] Briot, Hadjeres, and Pachet. 2020. Deep Learning Techniques for Music Generation. Vol. 1. Springer.

[13] Brown et al. 2020. Language Models Are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. Curran Associates, Inc., 1877–1901.

[14] Brunner, Konrad, Wang, and Wattenhofer. 2018. MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications To Style Transfer. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[15] Bubeck et al. 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. arXiv:2303.12712.

[16] Carr. 2004. Music, Meaning, and Emotion. *The Journal of Aesthetics and Art Criticism*, 62, 3, 225–234.

[17] Chang, Lee, and Yang. 2021. Variable-length Music Score Infilling Via XLNet and Musically Specialized Positional Encoding. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[18] Chen, Huang, Hsiao, and Yang. 2020. Automatic Composition of Guitar Tabs By Transformers and Groove Modeling. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[19] Chen and Su. 2021. Attend To Chords: Improving Harmonic Analysis of Symbolic Music Using Transformer-based Models. *Transactions of the International Society for Music Information Retrieval*, 4, 1, 1–13.

[20] Chen and Su. 2019. Harmony Transformer: Incorporating Chord Segmentation into Harmony Recognition. In *International Society for Music Information Retrieval Conference (ISMIR)* (Delft, The Netherlands). ISMIR, (Nov. 2019), 259–267.

[21] Choi, Park, Heo, Jeon, and Park. 2021. Chord Conditioned Melody Generation With Transformer Based Decoders. *IEEE Access*, 9, 42071–42080.

[22] Chomsky. 1980. Human Language and Other Semiotic Systems, 429–440.

[23] Chou, Chen, Chang, Ching, and Yang. 2021. MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding. arXiv:2107.05223.

[24] Chuan, Agres, and Herremans. 2020. From Context To Concept: Exploring Semantic Relationships in Music with Word2Vec. *Neural Computing and Applications*, 32, 1023–1036.

[25] Chung, Ahn, and Bengio. 2017. Hierarchical Multiscale Recurrent Neural Networks. In *International Conference on Learning Representations*.

[26] Chung, Gulcehre, Cho, and Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks On Sequence Modeling. In *NIPS 2014 Workshop on Deep Learning*.

[27] Cilibrasi, VitÁnyi, and Wolf. 2004. Algorithmic Clustering of Music Based On String Compression. *Computer Music Journal*, 28, 4, (Dec. 2004), 49–67.

[28] Conklin and Witten. 1995. Multiple Viewpoint Systems for Music Prediction. *Journal of New Music Research*, 24, 1, 51–73.

[29] Conneau, Kruszewski, Lample, Barrault, and Baroni. 2018. What You Can Cram Into a Single $&!#* Vector: Probing Sentence Embeddings for Linguistic Properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, (July 2018), 2126–2136.

[30] Cooke. 1959. The Language of Music.

[31] Cornelissen, Zuidema, Burgoyne, et al. 2020. Mode Classification and Natural Units in Plainchant. In *International Society for Music Information Retrieval Conference (ISMIR)*, 869–875.

[32] Corrêa and Rodrigues. 2016. A Survey On Symbolic Data-based Music Genre Classification. *Expert Systems with Applications*, 60, 190–210.

[33] Crocker. 1966. A History of Musical Style. *McGraw-Hill series in music*. McGraw-Hill Book Company.

[34] Dai, Jin, Gomes, and Dannenberg. 2021. Controllable Deep Melody Generation Via Hierarchical Music Structure Representation. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[35] Dai, Yang, Yang, Carbonell, Le, and Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, (July 2019), 2978–2988.

[36] Dalmazzo, Déguernel, and Sturm. 2023. The Chordinator: Modeling Music Harmony By Implementing Transformer Networks and Token Strategies.

[37] Dash and Agres. 2023. AI-Based Affective Music Generation Systems: A Review of Methods, and Challenges. arXiv:2301.06890.

[38] Dempster. 1998. Is There Even a Grammar of Music ? *Musicae Scientiae*, 2, 1, 55–65.

[39] Devlin, Chang, Lee, and Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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, (June 2019), 4171–4186.

[40] Di, Jiang, Liu, Wang, Zhu, He, Liu, and Yan. 2021. Video Background Music Generation with Controllable Music Transformer. In *Proceedings of the 29th ACM International Conference on Multimedia* (MM '21). Association for Computing Machinery, Virtual Event, China, 2037–2045.

[41] Domingo, García-Martínez, Helle, Casacuberta, and Herranz. 2023. How Much Does Tokenization Affect Neural Machine Translation? In *Computational Linguistics and Intelligent Text Processing*. Springer Nature Switzerland, Cham, 545–554.

[42] Donahue, Lee, and Liang. 2020. Enabling Language Models To Fill in the Blanks. arXiv:2005.05339.

[43] Donahue, Mao, Li, Cottrell, and McAuley. 2019. LakhNES: Improving Multi-instrumental Music Generation with Cross-domain Pre-training. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[44] Dong, Chen, Dubnov, McAuley, and Berg-Kirkpatrick. 2023. Multitrack Music Transformer. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

[45] Dong, Xu, and Xu. 2018. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5884–5888.

[46] Dosovitskiy et al. 2020. An Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale. In *International Conference on Learning Representations (ICLR)*.

[47] Douwes, Bindi, Caillon, Esling, and Briot. 2023. Is Quality Enough? Integrating Energy Consumption in a Large-Scale Evaluation of Neural Audio Synthesis Models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

[48] Downie. 1999. Evaluating a Simple Approach To Music Information Retrieval: Conceiving Melodic N-grams As Text. Faculty of Graduate Studies, University of Western Ontario London, Ont.

[49] Duan, Yu, Zhang, Tang, Li, and Oyama. 2023. Melody Generation From Lyrics with Local Interpretability. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19, 3, Article 124, (Feb. 2023), 21 pages.

[50] Ens and Pasquier. 2020. MMM : Exploring Conditional Multi-Track Music Generation with the Transformer. arXiv:2008.06048.

[51] Fernández and Vico. 2013. AI Methods in Algorithmic Composition: A Comprehensive Survey. *Journal of Artificial Intelligence Research*, 48, 513–582.

[52] Ferreira, Limongi, and Fávero. 2023. Generating Music with Data: Application of Deep Learning Models for Symbolic Music Composition. *Applied Sciences*, 13, 7.

[53] Fornäs. 1997. Text and Music Revisited. *Theory, Culture & Society*, 14, 3, 109–123.

[54] Fradet, Briot, Chhel, El Fallah-Seghrouchni, and Gutowski. 2021. MidiTok: A Python Package for MIDI File Tokenization. In *International Society for Music Information Retrieval Conference (ISMIR), Late-Breaking Demo Session.*

[55] Fradet, Gutowski, Chhel, and Briot. 2023. Byte Pair Encoding for Symbolic Music. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, (Dec. 2023), 2001–2020.

[56] Fradet, Gutowski, Chhel, and Briot. 2023. Impact of Time and Note Duration Tokenizations On Deep Learning Symbolic Music Modeling. In *International Society for Music Information Retrieval Conference (ISMIR).*

[57] Fu, Tanimura, and Nakada. 2023. Improve Symbolic Music Pre-training Model Using MusicTransformer Structure. In *2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 1–6.

[58] Gage. 1994. A New Algorithm for Data Compression. *C Users Journal*, 12, 2, 23–38.

[59] Garcia-Valencia. 2020. Embeddings As Representation for Symbolic Music. arXiv:2005.09406.

[60] Gardner, Grus, Neumann, Tafjord, Dasigi, Liu, Peters, Schmitz, and Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. arXiv:1803.07640.

[61] Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (NIPS'14). MIT Press, Montreal, Canada, 2672–2680.

[62] Gou, Yu, Maybank, and Tao. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129, 1789–1819.

[63] Gover and Zewi. 2022. Music Translation: Generating Piano Arrangements in Different Playing Levels. In *International Society for Music Information Retrieval Conference (ISMIR).*

[64] Goyal, Choudhury, Raje, Chakaravarthy, Sabharwal, and Verma. 2020. PoWER-BERT: Accelerating BERT Inference Via Progressive Word-vector Elimination. In *International Conference on Machine Learning*. PMLR, 3690–3699.

[65] Guan, Zhao, Qiu, Zhang, and Xia. 2018. Melodic Phrase Segmentation By Deep Neural Networks. arXiv:1811.05688.

[66] Guo, Simpson, Kiefer, Magnusson, and Herremans. 2022. MusIAC: An Extensible Generative Framework for Music Infilling Applications with Multi-level Control. In *Artificial Intelligence in Music, Sound, Art and Design*. Springer International Publishing, Cham, 341–356.

[67] Guo, Kang, and Herremans. 2023. A Domain-knowledge-inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence* (AAAI'23/IAAI'23/EAAI'23) Article 566. AAAI Press, 8 pages.

[68] Hadjeres and Crestel. 2021. The Piano Inpainting Application. arXiv:2107.05944.

[69] Hadjeres, Pachet, and Nielsen. 2017. DeepBach: a Steerable Model for Bach Chorales Generation. In *International conference on machine learning*. PMLR, 1362–1371.

[70] Han, Ihm, and Lim. 2023. Systematic Analysis of Music Representations From BERT. arXiv:2306.04628.

[71] Hentschel, Neuwirth, and Rohrmeier. 2021. The Annotated Mozart Sonatas: Score, Harmony, and Cadence. *Transactions of the International Society for Music Information Retrieval*, (May 2021).

[72] Hernandez-Olivan and Beltran. 2023. Musicaiz: A Python Library for Symbolic Music Generation, Analysis and Visualization. *SoftwareX*, 22, 101365.

[73] Herremans and Chuan. 2017. Modeling Musical Context with Word2vec. In *Proceedings of the International Workshop on Deep Learning and Music.*

[74] Herremans, Chuan, and Chew. 2017. A Functional Taxonomy of Music Generation Systems. *ACM Comput. Surv.*, 50, 5, Article 69, (Sept. 2017), 30 pages.

[75] Hillewaere, Manderick, and Conklin. 2018. Global Feature Versus Event Models for Folk Song Classification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*. ISMIR, Kobe, Japan, (Sept. 2018), 729–734.

[76] Hirai and Sawada. 2019. Melody2vec: Distributed Representations of Melodic Phrases Based On Melody Segmentation. *Journal of Information Processing*, 27, 278–286.

[77] Hochreiter and Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9, 8, (Nov. 1997), 1735–1780.

[78] Hsiao, Liu, Yeh, and Yang. 2021. Compound Word Transformer: Learning To Compose Full-song Music Over Dynamic Directed Hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35, 178–186.

[79] Huang, Dinculescu, Vaswani, and Eck. 2018. Visualizing Music Self-attention. In *Proc. NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language*, 1.

[80] Huang, Cooijmans, Roberts, Courville, and Eck. 2017. Counterpoint By Convolution. In *International Society for Music Information Retrieval Conference (ISMIR).*

[81] Huang, Duvenaud, and Gajos. 2016. ChordRipple: Recommending Chords To Help Novice Composers Go Beyond the Ordinary. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (IUI '16). Association for Computing Machinery, Sonoma, California, USA, 241–250.

[82] Huang et al. 2019. Music Transformer. In *International Conference on Learning Representations (ICLR).*

[83] Huang and Xue. 2012. Words Without Boundaries: Computational Approaches To Chinese Word Segmentation. *Language and Linguistics Compass*, 6, 8, 494–505.

[84] Huang, Sun, and Chen. 2010. Classical Chinese Sentence Segmentation. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*.

[85] Huang and Yang. 2020. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In *Proceedings of the 28th ACM International Conference on Multimedia* (MM '20). Association for Computing Machinery, Seattle, WA, USA, 1180–1188.

[86] Hung, Ching, Doh, Kim, Nam, and Yang. 2021. EMOPIA: A Multi-Modal Pop Piano Dataset For Emotion Recognition and Emotion-based Music Generation. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, Online, (Oct. 2021), 318–325.

[87] Jackendoff. 2009. Parallels and Nonparallels Between Language and Music. *Music Perception: An Interdisciplinary Journal*, 26, 3, 195–204.

[88] Jauhiainen, Lui, Zampieri, Baldwin, and Lindén. 2019. Automatic Language Identification in Texts: a Survey. *J. Artif. Int. Res.*, 65, 1, (May 2019), 675–682.

[89] Jeong, Kwon, Kim, Lee, and Nam. 2019. VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, Delft, The Netherlands, (Nov. 2019), 908–915.

[90] Ji, Yang, and Luo. 2023. A Survey On Deep Learning for Symbolic Music Generation: Representations, Algorithms, Evaluations, and Challenges. *ACM Comput. Surv.*, 56, 1, Article 7, (Aug. 2023), 39 pages.

[91] Jiang et al. 2023. Mistral 7B. arXiv:2310.06825.

[92] Jiang, Xia, and Berg-Kirkpatrick. 2020. Discovering Music Relations with Sequential Attention. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*. Association for Computational Linguistics, Online, (Oct. 2020), 1–5.

[93] Jiang, Xia, Carlton, Anderson, and Miyakawa. 2020. Transformer VAE: A Hierarchical Model for Structure-Aware and Interpretable Music Representation Learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 516–520.

[94] Jin, Jin, Hu, Vechtomova, and Mihalcea. 2022. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48, 1, (Mar. 2022), 155–205.

[95] Ju et al. 2022. TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method. arXiv:2109.09617.

[96] Jurafsky. 2000. Speech & Language Processing.

[97] Kang, Poria, and Herremans. 2023. Video2Music: Suitable Music Generation From Videos Using an Affective Multimodal Transformer Model. arXiv:2311.00968.

[98] Katharopoulos, Vyas, Pappas, and Fleuret. 2020. Transformers Are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proceedings of the 37th International Conference on Machine Learning* (ICML'20) Article 478. JMLR.org, 10 pages.

[99] Kermarec, Bigo, and Keller. 2022. Improving Tokenization Expressiveness With Pitch Intervals. In *International Society for Music Information Retrieval Conference (ISMIR), Late-Breaking Demo Session*.

[100] Kessler, Nunberg, and Schuetze. 1997. Automatic Detection of Text Genre. arXiv:cmp-lg/9707002.

[101] Kim et al. 2019. Probing What Different NLP Tasks Teach Machines About Function Word Comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 235–249.

[102] Kingma and Welling. 2013. Auto-encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*.

[103] Klein and Jacobsen. 2012. Music Is Not a Language: Re-interpreting Empirical Evidence of Musical 'Syntax'.

[104] Krausz. 2019. Rightness and Reasons: Interpretation in Cultural Practices. Cornell University Press.

[105] Kregor. 2015. Program Music. Cambridge University Press.

[106] Krol, Llano, and McCormack. 2022. Towards the Generation of Musical Explanations with GPT-3. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 131–147.

[107] Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, (July 2018), 66–75.

[108] Kumar and Sarmento. 2023. From Words To Music: A Study of Subword Tokenization Techniques in Symbolic Music Generation. arXiv:2304.08953.

[109] Lahnala, Kambhatla, Peng, Whitehead, Minnehan, Guldan, Kummerfeld, Çamcı, and Mihalcea. 2021. Chord Embeddings: Analyzing What They Capture and Their Role for Next Chord Prediction and Artist Attribute Prediction. In *Artificial Intelligence in Music, Sound, Art and Design*. Springer International Publishing, Cham, 171–186.

[110] Lazzari, Poltronieri, and Presutti. 2023. Pitchclass2vec: Symbolic Music Structure Segmentation with Chord Embeddings. arXiv:2303.15306.

[111] Lee, Kim, Kang, Ki, Hwang, Han, Kim, et al. 2022. ComMU: Dataset for Combinatorial Music Generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 39103–39114.

[112] Lemström and Pienimäki. 2007. On Comparing Edit Distance and Geometric Frameworks in Content-based Retrieval of Symbolically Encoded Polyphonic Music. *Musicae Scientiae*, 11, 1_suppl, 135–152. eprint: https://doi.org/10.1177/102986490701100106.

[113] Lerdahl. 2012. Musical Syntax and Its Relation To Linguistic Syntax. Collège de France.

[114] Lerdahl and Jackendoff. 1996. A Generative Theory of Tonal Music. MIT press, Cambridge, Massachusetts, USA.

[115] Lewis, Liu, Goyal, Ghazvininejad, Mohamed, Levy, Stoyanov, and Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461.

[116] Li and Sung. 2023. MelodyDiffusion: Chord-Conditioned Melody Generation Using a Transformer-Based Diffusion Model. *Mathematics*, 11, 8.

[117] Li and Sung. 2023. MRBERT: Pre-Training of Melody and Rhythm for Automatic Music Generation. *Mathematics*, 11, 4, 798.

[118] Li and Sung. 2023. Transformer-Based Seq2Seq Model for Chord Progression Generation. *Mathematics*, 11, 5.

[119] Li and Yang. 2018. Word Embedding for Understanding Natural Language: A Survey. In *Guide to Big Data Applications*. Springer International Publishing, Cham, 83–104.

[120] Li, Li, and Fazekas. 2023. An Comparative Analysis of Different Pitch and Metrical Grid Encoding Methods in the Task of Sequential Music Generation. arXiv:2301.13383.

[121] Li, Li, and Fazekas. 2023. Pitch Class and Octave-Based Pitch Embedding Training Strategies for Symbolic Music Generation. In *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research (CMMR)*. Zenodo, Tokyo, Japan, (Nov. 2023), 86–97.

[122] Liang, Lei, Chan, Yang, Sun, and Chua. 2020. PiRhDy: Learning Pitch-, Rhythm-, and Dynamics-aware Embeddings for Symbolic Music. In *Proceedings of the 28th ACM International Conference on Multimedia* (MM '20). Association for Computing Machinery, Seattle, WA, USA, 574–582.

[123] Liang et al. 2020. XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, (Nov. 2020), 6008–6018.

[124] Lidov. 1997. Our Time with the Druids: What (And How) We Can Recuperate From Our Obsession with Segmental Hierarchies and Other "Tree Structures". *Contemporary Music Review*, 16, 4, 1–28.

[125] Lin, Joty, Jwalapuram, and Bari. 2019. A Unified Linear-Time Framework for Sentence-Level Discourse Parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, (July 2019), 4190–4200.

[126] Liu and Ting. 2017. Computational Intelligence in Music Composition: A Survey. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1, 1, 2–15.

[127] Liu, Dong, Cheng, Zhang, Li, Yu, and Sun. 2022. Symphony Generation with Permutation Invariant Language Model. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[128] Liu, Kusner, and Blunsom. 2020. A Survey On Contextual Embeddings. arXiv:2003.07278.

[129] Liu et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

[130] Liutkus, Cífka, Wu, Simsekli, Yang, and Richard. 2021. Relative Positional Encoding for Transformers with Linear Complexity. In *Proceedings of the 38th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 139. PMLR, (July 2021), 7067–7079.

[131] Loiseau, Keller, and Bigo. 2021. What Musical Knowledge Does Self-Attention Learn ? In *Proceedings of the 2nd Workshop on NLP for Music and Spoken Audio (NLP4MusA)*. Association for Computational Linguistics, Online, (Nov. 2021), 6–10.

[132] Lu, Xu, Kang, Yu, Xing, Tan, and Bian. 2023. MuseCoco: Generating Symbolic Music From Text. *arXiv preprint arXiv:2306.00110*.

[133] Madjiheurem, Qu, and Walder. 2016. Chord2vec: Learning Musical Chord Embeddings. In *Proceedings of the constructive machine learning workshop at 30th conference on neural information processing systems (NIPS2016), Barcelona, Spain*.

[134] Makris, Agres, and Herremans. 2021. Generating Lead Sheets with Affect: A Novel Conditional Seq2seq Framework. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

[135] Makris, Zixun, Kaliakatsos-Papakostas, and Herremans. 2022. Conditional Drums Generation Using Compound Word Representations. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 179–194.

[136] McClary. 2002. Feminine Endings: Music, Gender, and Sexuality. U of Minnesota Press.

[137] McKay, Cumming, and Fujinaga. 2018. JSymbolic 2.2: Extracting Features From Symbolic Music for Use in Musicological and MIR Research. In *International Society for Music Information Retrieval Conference (ISMIR)*, 348–354.

[138] Merriam and Merriam. 1964. The Anthropology of Music. Northwestern University Press.

[139] Michel, Levy, and Neubig. 2019. Are Sixteen Heads Really Better Than One? 32.

[140] Mielke et al. 2021. Between Words and Characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. arXiv:2112.10508.

[141] Mikolov, Chen, Corrado, and Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781.

[142] Min, Jiang, Xia, and Zhao. 2023. Polyffusion: A Diffusion Model for Polyphonic Score Generation with Internal and External Controls. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[143] Muhamed, Li, Shi, Yaddanapudi, Chi, Jackson, Suresh, Lipton, and Smola. 2021. Symbolic Music Generation with Transformer-GANs. In number 1. Vol. 35. (May 2021), 408–417.

[144] Narmour. 1990. The Analysis and Cognition of Basic Melodic Structures: The Implication-realization Model. University of Chicago Press.

[145] Neves, Fornari, and Florindo. 2022. Generating Music with Sentiment Using Transformer-GANs. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[146] Noh. 2021. Analysis of Gradient Vanishing of RNNs and Performance Comparison. *Information*, 12, 11.

[147] Ogihara and Li. 2008. N-Gram Chord Profiles for Composer Style Representation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 671–676.

[148] Oore, Simon, Dieleman, Eck, and Simonyan. 2018. This Time with Feeling: Learning Expressive Musical Performance. *Neural Computing and Applications*, 32, 955–967.

[149] Ott, Edunov, Baevski, Fan, Gross, Ng, Grangier, and Auli. 2019. Fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 48–53.

[150] Ozaki, de Heer Kloots, Ravignani, and Savage. 2023. Cultural Evolution of Music and Language.

[151] Palmer. 2000. Tokenisation and Sentence Segmentation. *Handbook of natural language processing*, 11.

[152] Payne. 2019. Musenet. *OpenAI Blog*.

[153] Pearce. 2018. Statistical Learning and Probabilistic Prediction in Music Cognition: Mechanisms of Stylistic Enculturation. *Annals of the New York Academy of Sciences*, 1423, 1, 378–395.

[154] Peng, Yan, and Lu. 2019. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo On Ten Benchmarking Datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*. Association for Computational Linguistics, Florence, Italy, (Aug. 2019), 58–65.

[155] Perotti and Billoni. 2020. On the Emergence of Zipf's Law in Music. *Physica A: Statistical Mechanics and its Applications*, 549, 124309.

[156] Pollastri and Simoncelli. 2001. Classification of Melodies By Composer with Hidden Markov Models. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*, 88–95.

[157] Pulvermüller and Assadollahi. 2007. Grammar or Serial Order?: Discrete Combinatorial Brain Mechanisms Reflected By the Syntactic Mismatch Negativity. *Journal of cognitive neuroscience*, 19, 6, 971–980.

[158] Qin, Xie, Ding, Tan, Li, Zhao, and Ye. 2022. Bar Transformer: a Hierarchical Model for Learning Long-term Structure and Generating Impressive Pop Music. *Applied Intelligence*, 53, 9, (Aug. 2022), 10130–10148.

[159] Qiu, Li, and Sung. 2021. DBTMPE: Deep Bidirectional Transformers-Based Masked Predictive Encoder Approach for Music Genre Classification. *Mathematics*, 9, 5.

[160] Radford, Narasimhan, Salimans, Sutskever, et al. 2018. Improving Language Understanding By Generative Pre-training. *OpenAI*.

[161] Radford, Wu, Child, Luan, Amodei, and Sutskever. 2019. Language Models Are Unsupervised Multitask Learners.

[162] Ramesh, Pavlov, Goh, Gray, Voss, Radford, Chen, and Sutskever. 2021. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 139. PMLR, (July 2021), 8821–8831.

[163] Reimers and Gurevych. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. arXiv:1908.10084.

[164] Ren, He, Tan, Qin, Zhao, and Liu. 2020. Popmag: Pop Music Accompaniment Generation. In *Proceedings of the 28th ACM international conference on multimedia*, 1198–1206.

[165] Roberts, Engel, Raffel, Hawthorne, and Eck. 2018. A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In *Proceedings of the 35th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 80. PMLR, (July 2018), 4364–4373.

[166] Rohrmeier. 2011. Towards a Generative Syntax of Tonal Harmony. *Journal of Mathematics and Music*, 5, 1, 35–53.

[167] Sanh, Debut, Chaumond, and Wolf. 2020. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv:1910.01108.

[168] Sarmento, Kumar, Carr, Zukowski, Barthet, and Yang. 2021. DadaGP: A Dataset of Tokenized GuitarPro Songs for Sequence Models. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[169] Sarmento, Kumar, Chen, Carr, Zukowski, and Barthet. 2023. GTR-CTRL: Instrument And Genre Conditioning For Guitar-Focused Music Generation With Transformers. In *Artificial Intelligence in Music, Sound, Art and Design*. Springer Nature Switzerland, Cham, 260–275.

[170] Sarmento, Kumar, Xie, Carr, Zukowski, and Barthet. 2023. ShredGP: Guitarist Style-Conditioned Tablature Generation with Transformers, (Nov. 2023), 112–121.

[171] Schuster and Nakajima. 2012. Japanese and Korean Voice Search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5149–5152.

[172] Sears, Arzt, Frostel, Sonnleitner, and Widmer. 2017. Modeling Harmony with Skip-grams. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[173] Sennrich, Haddow, and Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, (Aug. 2016), 1715–1725.

[174] Serra-Peralta, Serrà, and Corral. 2021. Heaps' Law and Vocabulary Richness in the History of Classical Music Harmony. *EPJ Data Science*, 10, 1, 40.

[175] Shen, Yang, Yang, and Lin. 2023. More Than Simply Masking: Exploring Pre-training Strategies for Symbolic Music Understanding. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval* (ICMR '23). Association for Computing Machinery, Thessaloniki, Greece, 540–544.

[176] Shih, Wu, Zalkow, Müller, and Yang. 2023. Theme Transformer: Symbolic Music Generation With Theme-Conditioned Transformer. *IEEE Transactions on Multimedia*, 25, 3495–3508.

[177] Stamatatos. 2009. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology*, 60, 3, 538–556.

[178] Sturm, Santos, Ben-Tal, and Korshunova. 2016. Music Transcription Modelling and Composition Using Deep Learning. arXiv:1604.08723.

[179] Sulun, Davies, and Viana. 2022. Symbolic Music Generation Conditioned On Continuous-Valued Emotions. *IEEE Access*, 10, 44617–44626.

[180] Tang, Wiggins, and Fazekas. 2023. Reconstructing Human Expressiveness in Piano Performances with a Transformer Network. arXiv:2306.06040.

[181] Tao, Tong, Zhao, Xu, Jin, and Liu. 2019. A Radical-Aware Attention-Based Model for Chinese Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 01, (July 2019), 5125–5132.

[182] Trieu and Keller. 2018. JazzGAN: Improvising with Generative Adversarial Networks. In *MUME workshop*.

[183] Van Der Merwe and Schulze. 2011. Music Generation with Markov Models. *IEEE MultiMedia*, 18, 3, 78–85.

[184] Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin. 2017. Attention Is All You Need. 30.

[185] Vercoe. 2001. Folk Music Classification Using Hidden Markov Models. In *Proceedings of the International Conference on Artificial Intelligence*. Vol. 6.

[186] Voita, Talbot, Moiseev, Sennrich, and Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, (July 2019), 5797–5808.

[187] Von Rütte, Biggio, Kilcher, and Hofmann. 2022. FIGARO: Generating Symbolic Music with Fine-Grained Artistic Control. arXiv:2201.10936.

[188] Wallin, Merker, and Brown. 2001. The Origins of Music. MIT press.

[189] Wang, Singh, Michael, Hill, Levy, and Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Brussels, Belgium, (Nov. 2018), 353–355.

[190] Wang, Wang, Chen, Wang, and Kuo. 2019. Evaluating Word Embedding Models: Methods and Experimental Results. *APSIPA Transactions on Signal and Information Processing*, 8, 19.

[191] Wang, Zhao, Liu, Pang, Qin, and Wu. 2023. A Review of Intelligent Music Generation Systems. arXiv:2211.09124.

[192] Wang and Xia. 2021. MuseBERT: Pre-training of Music Representation for Music Understanding and Controllable Generation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 722–729.

[193] Wang, Zhang, Zhang, Jiang, Yang, Xia, and Zhao. 2020. PianoTree VAE: Structured representation learning for polyphonic music. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, Montreal, Canada, (Nov. 2020), 368–375.

[194] Wankhade, Rao, and Kulkarni. 2022. A Survey On Sentiment Analysis Methods, Applications, and Challenges. *Artificial Intelligence Review*, 55, 7, 5731–5780.

[195] Weizenbaum. 1966. ELIZA—a Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM*, 9, 1, (Jan. 1966), 36–45.

[196] Wolf et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, (Oct. 2020), 38–45.

[197] Wołkowicz and Kešelj. 2012. Analysis of Important Factors for Measuring Similarity of Symbolic Music Using N-gram-Based, Bag-of-Words Approach. In *Advances in Artificial Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 230–241.

[198] Wołkowicz, Kulka, and Kešelj. 2008. N-gram-based Approach To Composer Recognition. *Archives of Acoustics*, 33, 1, 43–55.

[199] Wong, Li, Xu, and Zhang. 2022. Introduction To Chinese Natural Language Processing. Springer Nature.

[200] Wu and Sun. 2023. Exploring the Efficacy of Pre-trained Checkpoints in Text-to-Music Generation Task. In *The AAAI-23 Workshop on Creative AI Across Modalities*.

[201] Wu, Yu, Tan, and Sun. 2023. CLaMP: Contrastive Language-Music Pre-training for Cross-Modal Symbolic Music Information Retrieval. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[202] Wu and Yang. 2023. Compose & Embellish: Well-Structured Piano Performance Generation Via A Two-Stage Approach. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

[203] Wu and Yang. 2023. MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer With One Transformer VAE. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31, 1953–1967.

[204] Wu and Yang. 2020. The Jazz Transformer On the Front Line: Exploring the Shortcomings of AI-composed Music Through Quantitative Measures. In *International Society for Music Information Retrieval Conference (ISMIR)*, 142–149.

[205] Wu, Wang, and Lei. 2020. Transformer-XL Based Music Generation with Multiple Sequences of Time-valued Notes. arXiv:2007.07244.

[206] Xu, Wang, Tian, Xu, Zhao, Wang, and Hao. 2015. Short Text Clustering Via Convolutional Neural Networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Association for Computational Linguistics, Denver, Colorado, (June 2015), 62–69.

[207] Yang, Dai, Yang, Carbonell, Salakhutdinov, and Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32. Curran Associates, Inc., Vancouver, Canada.

[208] Yannakakis and Martínez. 2015. Ratings Are Overrated! *Frontiers in ICT*, 2.

[209] Yu, Lu, Wang, Hu, Tan, Ye, Zhang, Qin, and Liu. 2022. Museformer: Transformer with Fine-and Coarse-grained Attention for Music Generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 1376–1388.

[210] Zbikowski. 2009. Music, Language, and Multimodal Metaphor. *Multimodal metaphor*, 359–381.

[211] Zeng, Tan, Wang, Ju, Qin, and Liu. 2021. MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, (Aug. 2021), 791–800.

[212] Zhang, Karystinaios, Dixon, Widmer, and Cancino-Chacón. 2023. Symbolic Music Representations for Classification Tasks: A Systematic Evaluation. In *International Society for Music Information Retrieval Conference (ISMIR)*.

[213] Zhang and Callison-Burch. 2023. Language Models Are Drummers: Drum Composition with Natural Language Pre-Training. In *The AAAI-23 Workshop on Creative AI Across Modalities*.

[214] Zhang and Jiang. 2021. Visualizing Symbolic Music Via Textualization: An Empirical Study On Chinese Traditional Folk Music. In *Mobile Multimedia Communications*. Springer International Publishing, Cham, 647–662.

[215] Zhang. 2020. Learning Adversarial Transformer for Symbolic Music Generation. *IEEE Transactions on Neural Networks and Learning Systems*, 34, 4, 1754–1763.

[216] Zhao, Chen, Yang, Liu, Deng, Cai, Wang, Yin, and Du. 2024. Explainability for Large Language Models: A Survey. *ACM Trans. Intell. Syst. Technol.*, (Jan. 2024).

[217] Zhao, Taniar, Adhinugraha, Baskaran, and Wong. 2023. Multi-mmlg: a Novel Framework of Extracting Multiple Main Melodies From MIDI Files. *Neural Computing and Applications*, 35, 30, 22687–22704.

[218] Zhao, Wong, Baskaran, Adhinugraha, and Taniar. 2023. Computational Music: Analysis of Music Forms. In *Computational Science and Its Applications – ICCSA 2023: 23rd International Conference, Athens, Greece, July 3–6, 2023, Proceedings, Part I*. Springer-Verlag, Athens, Greece, 366–384.

[219] Zhao, Xia, and Wang. 2023. AccoMontage-3: Full-Band Accompaniment Arrangement Via Sequential Style Transfer and Multi-Track Function Prior. arXiv:2310.16334.

[220] Zhao, Xia, and Wang. 2023. Q&A: Query-Based Representation Learning for Multi-Track Symbolic Music Re-Arrangement. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*.

[221] Zhou, Zhu, and Wang. 2023. Choir Transformer: Generating Polyphonic Music with Relative Attention On Transformer. arXiv:2308.02531.

[222] Zhu et al. 2018. XiaoIce Band: A Melody and Arrangement Generation Framework for Pop Music. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (KDD '18). Association for Computing Machinery, London, United Kingdom, 2837–2846.

[223] Zhu, Li, Liu, Ma, and Wang. 2023. A Survey On Model Compression for Large Language Models. arXiv:2308.07633.

[224] Zhu, Baca, Rekabdar, and Rawassizadeh. 2023. A Survey of AI Music Generation Tools and Models. arXiv:2308.12982.

[225] Zixun, Makris, and Herremans. 2021. Hierarchical Recurrent Neural Networks for Conditional Melody Generation with Long-term Structure. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8.