Published: 21 February 2025
Online AM: 28 January 2025
Accepted: 06 January 2025
Revised: 15 October 2024
Received: 26 February 2024

SURVEY

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

DINH-VIET-TOAN LE, University of Lille, Lille, Hauts-de-France, France

LOUIS BIGO, University of Bordeaux, Bordeaux, Nouvelle-Aquitaine, France

DORIEN HERREMANS, Singapore University of Technology and Design, Singapore City, Singapore

MIKAELA KELLER, University of Lille, Lille, Hauts-de-France, France

Open Access Support provided by:

University of Bordeaux

Singapore University of Technology and Design

University of Lille

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, F-33400 Talence, France
DORIEN HERREMANS, Singapore University of Technology and Design, Singapore, Singapore
MIKAELA KELLER, Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

Music is frequently associated with the notion of language, as both domains share several similarities, including the ability for their content to be represented as sequences of symbols. In computer science, the fields of Natural Language Processing (NLP) and Music Information Retrieval (MIR) reflect this analogy through a variety of similar tasks, such as author detection or content generation. This similarity has long encouraged the adaptation of NLP methods to process musical data, particularly symbolic music data, and the rise of Transformer neural networks has considerably strengthened this practice.

This survey reviews NLP methods applied to symbolic music generation and information retrieval following two axes. We first propose an overview of representations of symbolic music inspired by text sequential representations. We then review a large set of computational models, particularly deep learning models, which have been adapted from NLP to process these musical representations for various MIR tasks. These models are described and categorized through different prisms with a highlight on their music-specialized mechanisms. We finally present a discussion surrounding the adequate use of NLP tools to process symbolic music data. This includes technical issues regarding NLP methods which may open several doors for further research into more effectively adapting NLP tools to symbolic MIR.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → *Sound and music computing*; • **Information systems** → *Music retrieval;*

Additional Key Words and Phrases: Music information retrieval, natural language processing, symbolic music, music generation, music analysis, deep learning

ACM Reference Format:
Dinh-Viet-Toan Le, Louis Bigo, Dorien Herremans, and Mikaela Keller. 2025. Natural Language Processing Methods for Symbolic Music Generation and Information Retrieval: A Survey. *ACM Comput. Surv.* 57, 7, Article 175 (February 2025), 40 pages. https://doi.org/10.1145/3714457

This project received support from the Merlion PHC Music Language Processing N°48304SM funded by Campus France, the ANR-20-THIA-0014 program "AI_PhD@Lille," and SUTD's Kickstart Initiative under grant number SKI 2021_04_06.
Authors' Contact Information: Dinh-Viet-Toan Le (Corresponding author), Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France; email: dinhviettoan.le@univ-lille.fr; Louis Bigo, Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, F-33400 Talence, France; e-mail: louis.bigo@u-bordeaux.fr; Dorien Herremans, Singapore University of Technology and Design, Singapore, Singapore; e-mail: dorien_herremans@sutd.edu.sg; Mikaela Keller, Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France; e-mail: mikaela.keller@univ-lille.fr.

ACM 0360-0300/2025/02-ART175
https://doi.org/10.1145/3714457

1 Introduction and Background

The evolution of **Natural Language Processing (NLP)** has been marked by a substantial journey, progressing from rudimentary rule-based systems like ELIZA [230] 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 adopted NLP approaches to process 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.

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 allow computers to deal with human languages while taking into account their characteristics, such as syntactic or semantic properties which are essential for language understanding, interpretation, or generation. 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 MIR combines musicology and computer science to develop techniques for analyzing music or retrieving music-related data. It has been extended in recent years to encompass as well techniques for music generation. While audio files encode music as sound, at a low representation level such as waveforms or spectrograms, *symbolic music* consists of abstract notations representing concepts such as notes, chords, or intervals, which 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. The scope of this survey is limited to *symbolic* music 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 [103]. Both are specific to human species and are learned through imitation. Both can be deployed under two modalities: an annotated form (text, sheet music) and an auditory form (speech, musical performance) [62]. Several similarities can also be found from a structural point of view while specificities remain.

Hierarchical Representations. Text and symbolic music representations are both semiotic systems [27] based on arrangements of symbols. Text is built on characters or ideograms, and written music can be transcribed with a variety of symbols derived from various notation systems such as standard notation, numbered notation, or tablature. 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, sentences, and paragraphs, whereas music can typically be segmented into temporal units such as notes, motifs, musical phrases, or sections [129] as represented in Figure 1.

However, while white-spaces facilitate token segmentation of text in many languages, identifying boundaries of musical motives and phrases remain subjective [141] and can rise overlap problems [87]. In this sense, musical scores might more easily be compared to unsegmented languages [172] where word segmentation can be unclear [96].

Underlying Time and Simultaneity in Music. Text and music can be perceived as elements unfolding in time [250]. While speech might have a temporal dimension in terms of speech rate [223],

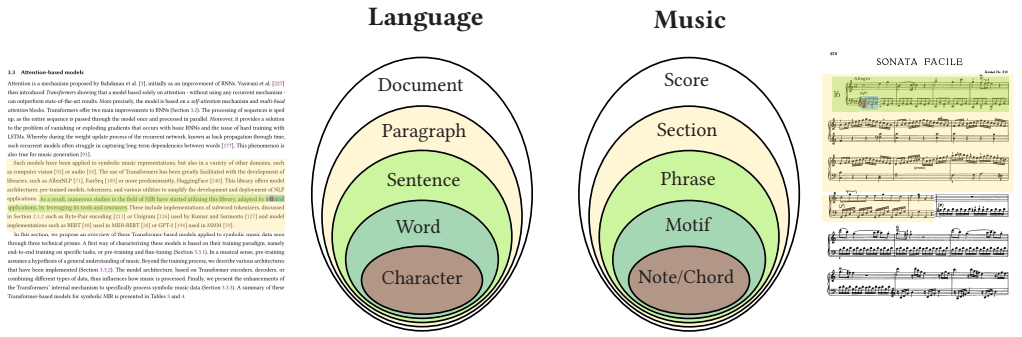

Fig. 1. Possible segmentation levels in text and symbolic music. Such segmentations can, however, include more or less fine-grained levels and their delimitations can be ambiguous (Section 4).

text does not explicitly encode any of these rhythmic modulations. In contrast, musical rhythm is based on an isochronic grid [103] in which notes are notated with rigorous timings, in terms of *onsets* and *durations*, beyond some microtimings linked to performance embellishments or tempo changes. In addition, while sequences 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]. As a consequence, slicing and tokenization methods have been elaborated to represent polyphonic music as sequences of elements, although this generally requires an approximation or complexification of the original data as presented in Section 2.

Symbol Polymorphism. The elements that constitute a musical score are less homogeneous than text data. While textual elements are of a single type (characters, ideograms with possibly punctuation), music symbols combine structural elements (bar, beat, etc.), note-related information (pitch, duration, dynamics, etc.), and global information (tempo, instrument, etc.).

Grammars. Inspired by higher-level concepts in natural language, multiple models of musical syntaxes have been proposed [6, 8]. Musical grammars rely on intrinsic concepts such as tension and relaxation [130], harmony [192], or the implication-realization mechanism [166]. Grammatical and syntactic rules induce expectancy in both language and music [103, 175], leading to similar cognitive reactions for the interlocutor or the listener when they are being transgressed in both fields [7, 179]. However, the existence of a global grammar describing music is not unanimously accepted, even in a specific style [43].

1.2 Applying NLP Methods in Symbolic MIR

Common Tasks in MIR and NLP. Beyond the preceding parallels between text and symbolic music representations, the NLP and MIR research fields are also related by similar tasks they address. On the one hand, tasks involving *labeled* data that aim to classify whole textual document or music piece are common, such as music composer classification [178] and text authorship attribution [207], folk song origin classification [91] and language detection [105], music genre [36] and text style [118] classification, or music emotion [102] and sentiment [229] 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 [78] or musical form analysis [258] in MIR and discourse parsing [142] or phrase segmentation [100] in NLP.

On the other hand, a variety of tasks rely on *unlabeled* music and text datasets. Apart from clustering tasks in text [242] and music [31], 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

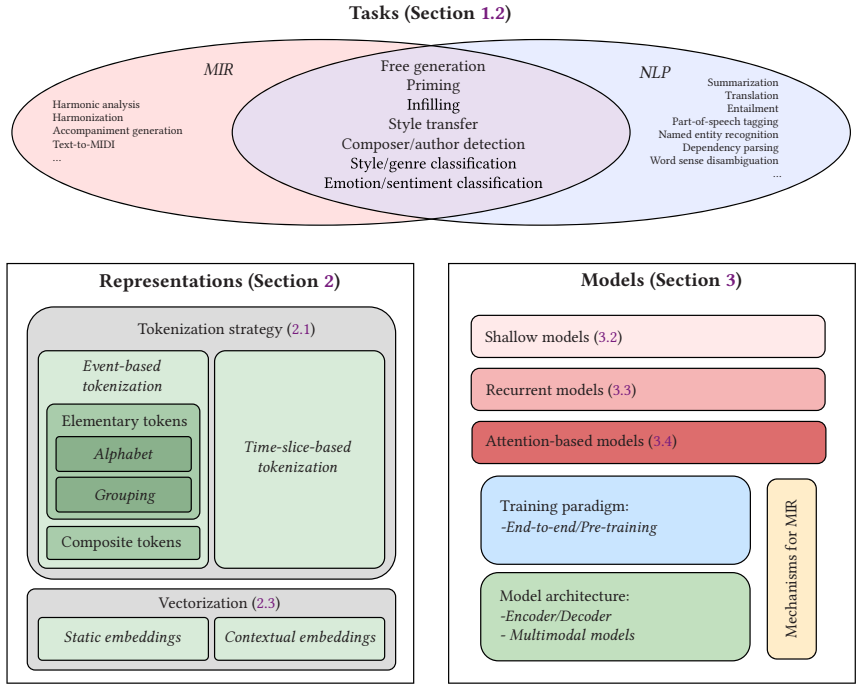

Fig. 2. Overview of the survey, organized around three axes: *similarities and specificities* of NLP and MIR tasks motivating *representations* of symbolic music inspired by NLP and *models* adapted from NLP for symbolic music.

as symbolic music infilling [79] and text infilling [50], or music priming [99] and text continuation [184]. At the scale of a piece or a document, style transfer is performed in both MIR, through musical genres [241], and NLP, through language high-level elements such as formality or toxicity [112]. 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 [150].

However, NLP and MIR also include numerous tasks that are inherent to one field, as depicted in Figure 2. These tasks specific to each field also reflect fundamental differences between these two types of data, including semantics in language which is crucial in an entailment task, or polyphony in music which is at the heart of harmonization and accompaniment generation tasks.

A variety of task evaluation methods have been implemented in both fields. In NLP, generative models are usually evaluated on benchmarks with a variety of metrics [17]. MIR generative models are usually evaluated through user studies, taking the form of preference selection [206], ranking [222], or scoring [151]. To counterbalance this subjective aspect, multiple quantitative music-related metrics have been proposed to evaluate music generation [244]. For an in-depth overview of music generation metrics, refer to the work of Ji et al. [107, p. 27].

Prominence of NLP Methods in Symbolic MIR. To address the preceding tasks, the MIR community has closely followed advances in NLP by adapting successful tools from this field. This seems particularly true for symbolic music generation for which multiple surveys have been published in the past decade. These surveys generally fall into two categories. A first category categorizes these generative systems from a technical perspective. These systems rely on methods such as

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

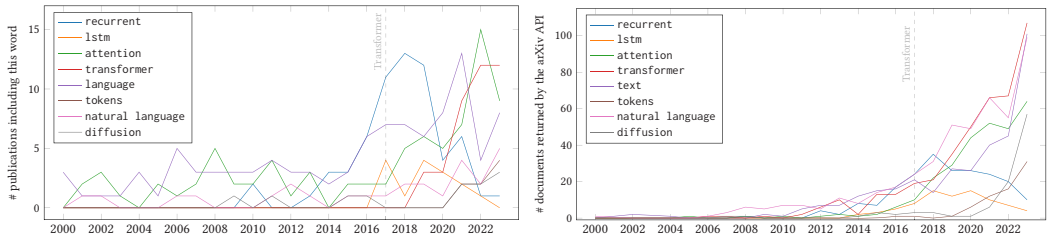

Fig. 3. Evolution of the number of articles containing NLP-related words. Left: Number of ISMIR papers containing NLP-related words in their abstracts from 2000 to 2023. Right: Number of arXiv preprints returned by the API query "music AND <term>".

grammars or Markov chains [60], or more recently on deep learning methods [12], organized by model architecture [226] or types of generation conditions [41, 265]. A second category of overviews bring together works that share a common musical purpose or task [90] and categorize them based on the nature of the generated content [143] or by the context surrounding the generated content [107].

Although MIR studies commonly adapt techniques from other fields such as image processing [97], a prominent amount of symbolic music generation approaches are adapted or inspired from NLP methods. Figure 3 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. The rise of Transformers from 2017 has largely contributed to increase these references, and a large number of the NLP-derived state-of-the-art models in symbolic MIR are now based on this model. This trend has encouraged dedicated initiatives in the MIR community, such as the organization of the workshop NLP4MuSA (NLP for Music and Spoken Audio).[2] 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]

Although not the focus of the present survey, it is finally worth noting that NLP methods have also widely been applied to audio [1, 34]. Such audio applications have gained popularity among the general public through commercial products, including audio music generation platforms[5] or AI-based audio effects,[6] which can then be integrated into a human-machine creative process [213].

1.3 Survey Outline

The original approach introduced in this survey emphasizes the adaptation of NLP methods for music generation and information retrieval within the domain of symbolic music. These encompass tools and methods not only for symbolic music generation, which constitutes a large part of MIR research today, but also for existing analysis tasks. From a more epistemological point of view, it is our 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.

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

We present an overview of NLP methods adapted for symbolic MIR by proposing taxonomies of two technical aspects presented next and also shown in Figure 2—*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 mainly involves 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 of neural networks, with specific architectures or training paradigms, and potentially mechanisms specifically tailored for symbolic music data. Although Transformers are only a limited part of NLP approaches, they appear to be by far the predominant NLP model used in MIR today. For this reason, this survey will mostly focus on attention-based models after mentions of shallow and recurrent models.

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

The MIR community frequently releases new models or methods adapted from NLP. This survey includes such developments up until mid-2024. A collaborative repository is maintained to facilitate continuous updates of newly released tools: 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 proposed to represent *music as sequences of elements*.

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* [161], such as subwords, words, or multiple-word expressions. The rise of NLP models in MIR has naturally encouraged the adoption of this term for music representations. We propose a taxonomy of tokenization strategies in symbolic MIR represented in Figure 2 (left).

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 commonly thought as occurring on an underlying isochronic grid [103] in which notes have rigorous timings annotated on sheet music.[7] Representing time properties of musical elements has led to multiple

[7]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*.

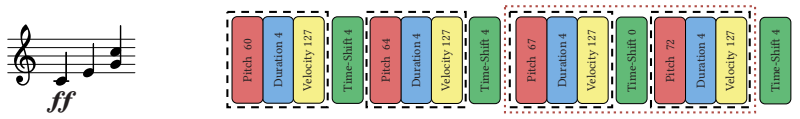

Fig. 4. Artificial sequentiality possibly introduced in a tokenization strategy (e.g., Structured [81]). With a note characterized by its pitch, duration, velocity, and time-shift, the sequentiality of the blocks (black dashed) follows the temporality, but the order of the inner musical features is arbitrary. The sequentiality of these blocks can even be artificial for simultaneous events (red dotted).

approaches [12, Section 4.8] including representations based on regular time steps (Section 2.1.1) or driven by events occurring through time (Section 2.1.2).

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 [83] is a model that aims to generate four-part chorales, for which time is evenly divided at the level of 16th notes. As the number of simultaneous notes is upper-bounded in four-part chorales, a time step can be represented as a vector containing four pitches. In the same way, a concept of "musical words" defined by slices of three beats is proposed [29, 89] to model musical context and semantic relationships in polyphonic music. Beyond pitches, this time-slice representation has also shown to be adapted to the context of drum music [253]. More generally, these representations can be seen as specific cases of *piano rolls*. A piano roll representation relies on a matrix in which the horizontal axis represents time, and pitches are encoded along the vertical axis, with possible 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—that is, 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 (a note being played, the start of a bar, etc.). Most tokenization strategies have shifted toward this event-centric approach, helped by the large 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 [107] which are essential for training deep learning models.

In contrast with characters in text, MIDI messages can have various types, reflecting the multiple features of musical notes 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 4. 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* (Section 2.1.2.1, Table 1) and *composite tokens* (Section 2.1.2.2, 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

Table 1. Overview[8] of Event-Based Tokenization Strategies Based on *Elementary* Tokens

Tokenization	Alphabet *(Atomic elements)*			Grouping	Vocab. size	Data
ABC notation [209]	Text alphabet			Bar patching [238]	N/A	Monophonic (Score)
SMT-ABC [182] *(MuPT)*	Text alphabet			BPE [182]	N/A	Multi-track (Score)
Park et al. [173] *(Mel2Word)*	`<Pitch-interval>` *(integer)*	`<Time-shift>` *(music time)*		BPE [173]	30	Monophonic (Score)
MIDI-like [170]	`<Note-ON>` *(MIDI value)* `<Time-shift>` *(absolute time)*	`<Note-OFF>` *(MIDI value)* `<Velocity>` *(integer)*		BPE [123, 252] Unigram [123]	388	Piano (Perf.)
LakhNES [51]	`<NoteON-[trk]>` *(MIDI value)*	`<NoteOFF-[trk]>` *(MIDI value)*	`<Time-shift>` *(absolute time)*	–	630	Multi-track (Perf.)
REMI [101]	`<Pitch>` *(MIDI value)* `<Chord>` *(class)*	`<Duration>` *(music time)* `<Bar>`	`<Velocity>` *(integer)* `<Position>` *(music time)*	BPE [64, 123, 252] Unigram [123]	332	Piano (Score)
REMI+ [222]	REMI alphabet + features: `<Time-Signature>` *(class)*	`<Instrument>` *(class)* `<Tempo>` *(integer)*		–	N/A	Multi-track (Score)
Lee et al. [128] *(ComMU)*	REMI alphabet + metadata: `<Instrument>` *(class)* `<BPM>` *(integer)* `<Pitch-range>` *(class)*	`<Key>` *(class)* `<Min/Max-velocity>` *(integer)* `<Rhythm>` *(class)*	`<Time-Signature>` *(class)* `<Nb-of-bars>` *(number)*	–	728	Multi-track (Score)
MusIAC [79]	REMI alphabet + control info: `<Tensile-train>` *(class)*	`<Occupation>` *(class)* `<Cloud diameter>` *(class)*	`<Density>` *(class)* `<Polyphony>` *(class)*	–	360	Multi-track (Score)
Wu and Yang [241] *(MuseMorphose)*	`<Durat.-[trk]>` *(music time)* `<Velocity-[trk]>` *(integer)*	`<Pitch-[trk]>` *(MIDI value)* `<Position>` *(music time)*	`<Bar>` `<Tempo>` *(integer)*	–	3440	Multi-track (Score)

MultiTrack [58]	`<Start-piece>`
	`<Start-bar>/<End-bar>`
	`<Note-ON/OFF>` *(MIDI value)*
		`<Start-track>/<End-track>`
		`<Start-fill>/<End-fill>`
		`<Time-shift>` *(absolute time)*
			`<Instrument>` *(class)*
			`<Density-level>` *(integer)*
				–	440	Multi-track (Perf.)

MMR [144] *(SymphonyNet)*	`<Start-score>/<End-score>` `<Position>` *(integer)* `<Chord>` *(class)*	`<Start-bar>/<End-bar>` `<Pitch>` *(MIDI value)*	`<Change-track>` `<Duration>` *(music time)*	BPE [144]	N/A	Multi-track (Score)
TSD [64]	`<Pitch>` *(MIDI value)* `<Duration>` *(absolute time)*	`<Velocity>` *(integer)* `<Time-shift>` *(absolute time)*	`<Rest>` *(absolute time)* `<Program>` *(class)*	BPE [64]	249	Multi-track (Perf.)
Structured [81]	`<Pitch>` *(MIDI value)* `<Duration>` *(absolute time)*	`<Velocity>` *(integer)* `<Time-shift>` *(absolute time)*		–	428	Piano (Perf.)
Li et al. [136]	`<Pitch-class>` *(class)* `<Bar>` *(integer)*	`<Octave>` *(integer)* `<Position>` *(music time)*	`<Duration>` *(music time)* `<Velocity>` *(integer)*	–	N/A	Monophonic (Score)
Chen et al. [22]	`<Pitch>` *(MIDI value)* `<Bar>` *(integer)* `<String>` *(integer)*	`<Duration>` *(music time)* `<Position>` *(music time)* `<Fret>` *(integer)*	`<Velocity>` *(integer)* `<Grooving>` *(class)* `<Technique>` *(class)*	–	231	Guitar (Tablatures)
DadaGP [195]	`<start>/<end>` `<Effect>` *(class)* `<String>` *(integer)*	`<Wait>` *(integer)* `<Drums:note>` *(MIDI value)* `<Fret>` *(integer)*	`<Instr:note>` *(MIDI value)*	BPE [123] Unigram [123]	2140	Guitar (Tablatures)

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. It also specified whether the tokenization is score or performance based.

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 a Sequence of Individual Features. The constitutive elements of a sequence composed of musical elementary tokens can be described at two levels (see 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* most often 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 (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.

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

Table 2. Overview[9] of Event-Based Tokenization Strategies Based on *Composite* Tokens

Tokenization	Musical features			Super-token nature	Data
PiRhDy [138]	<Chroma> *(class)* <Inter-onset-interval> *(music time)*	<Octave> *(integer)*	<Velocity> *(integer)* <Note-state> *(class)*	Homogeneous	Multi-track
Zixun et al. [266]	<Pitch> *(one-hot)* <Current-chord> *(one-hot)*	<Duration> *(one-hot)* <Next-chord> *(one-hot)*	<Bar> *(one-hot)*	Homogeneous	Lead sheet
Octuple [251]	<Time-signature> *(class)* <Position> *(music time)* <Velocity> *(integer)*	<Tempo> *(integer)* <Pitch> *(MIDI value)* <Instrument> *(class)*	<Bar> *(integer)* <Duration> *(music time)*	Homogeneous	Multi-track
Dong et al. [52] *(MMT)*	<Type> *(class)* <Pitch> *(MIDI value)*	<Beat> *(integer)* <Duration> *(music time)*	<Position> *(music time)* <Instrument> *(class)*	Homogeneous	Multi-track
Dalmazzo et al. [40] *(Chordinator)*	<Chord-root> *(class)* <Slash-chord> *(Boolean)*	<Chord-nature> *(class)* <MIDI-array> *(multi-hot)*	<Chord-extensions> *(class)*	Homogeneous	Chord sequences
Wang and Xia [228] *(MuseBERT)*	<Onset> *(music time)*	<Pitch> *(MIDI value)*	<Duration> *(music time)*	Homogeneous	Multi-track
MuMIDI [189]	<Bar> <Track> *(class)* <Velocity> *(integer)*	<Position> *(music time)* <Chord> *(class)* <Duration> *(music time)*	<Tempo> *(integer)* <Pitch/Drum> *(MIDI value)*	Family based	Multi-track
Compound Word [94]	<Family> *(class)* <Beat> *(music time)* <Pitch> *(MIDI value)*	<Time-signature> *(class)* <Chord> *(class)* <Duration> *(music time)*	<Bar> *(integer)* <Tempo> *(integer)* <Velocity> *(integer)*	Family based	Piano
Di et al. [48]	<Type> *(class)* <Pitch> *(MIDI value)*	<Beat> *(integer)* <Duration> *(music time)*	<Strenth> *(class)* <Instrument> *(integer)*	Family based	Multi-track
Makris et al. [154]	Encoder input: <Group> *(class)* Decoder output:	<Onset> *(number)* <Type> *(class)* <Onset> *(number)*	<Duration> *(music time or none)* <Value> *(any—depends on type)* <Drums> *(integer)*	Family based	Enc.: Multi-track Dec.: Drums
Unsupervised CPWord [214]	<Family> *(class)* <Beat> *(music time)* <Pitch> *(MIDI value)*	<Time-signature> *(class)* <Chord> *(class)* <Duration> *(music time)*	<Bar> *(integer)* <Tempo> *(integer)* <Velocity> *(integer)*	Family based + learning	Piano
REMI_Track [151]	<Instrument> *(class)* <BPE> | <Pitch> *(MIDI value)* | <Velocity> *(integer)* | <Duration> *(music time)*	<Position> *(music time)*	<Bar>	Heterogeneous + learning	Multi-track

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.

• *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.

• ***Building an* Alphabet *of Atomic Elements to Encode Music.*** Symbolic music alphabets first depend on whether the content is a "MIDI Score" or a "MIDI Performance" [170]. The first one is a MIDI file converted from a sheet music format (`musicXML`, `kern`...) exactly following a written metrical grid, whereas the second one is a performance encoded into the MIDI protocol. Scores include information such as exact musical timings and enharmonics, whereas performance data includes velocity and 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 [99] follows MIDI events from the basic MIDI protocol, including a vocabulary of four event types: `<note_on>`, `<note_off>`, `<time_shift>`, and `<velocity>`. This tokenization can be adapted for monophonic melodies [191] or a polyphonic ensemble with a fixed number of instruments [51] by having `<note_on/off>` tokens specific to each instrument. TSD (Time-Shift-Duration) [64] adapts the MIDI-like tokenization, using `<duration>` and `<time_shift>` to replace pairs of `<note_on/off>`. The Structured MIDI encoding [81] is similar to TSD but enforces the order of tokens describing a same event. This avoids syntax errors in the context of live music

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

generation and improves token sequence consistency by implicitly reducing the vocabulary size at each generation step.

In contrast, *score-based* tokenizations describe music as a time-structured system based on multiple discretization levels of time. REMI (Revamped MIDI-derived events) [101] 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 [138] and generative tasks [136]. Multiple extensions of REMI have been implemented, adding additional tokens including metadata [128], musical features [205, 222], control tokens [79], hand positioning for piano music [75], or track information [241]. Prior to MIDI-based tokenization, early sequential representations of music rely on the various score dimensions [32]. These representations, called *viewpoints*, can encode 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 [209] where tokens are considered to be text characters. Basic NLP models can be simply trained on these textual data for generation [209]. With the breakthrough of efficient **Large Language Models (LLMs)** handling text, this representation has been used for text-to-music systems such as ChatMusician [249] or MuPT [182] implementing a SMT-ABC (Synchronized Multi-Track ABC Notation) which improves bar and track consistency for multi-track music.

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 [22, 195] 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 events as a single sequence of tokens. The representations from MMM (Multi-track Music Machine) [58], MuMIDI [189], and the MMR (Multi-track Multi-instrument Repeatable) representation [144] 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, whereas MMR and MuMIDI have a vertical reading of the score by first concatenating simultaneous bars 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 to compress the sequence length by *grouping* tokens together.

A textual n-gram [114, 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 [56], then improved by n-grams/skip-grams [200]. 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 [234], musical descriptors [32], or chord n-grams to represent music through harmony [169]. These musical n-grams also show statistical phenomena

initially observed in text data representations. Various laws such as Heaps' law [202] or Zipf's law [177, 234] can be observed with musical n-grams. Musically informed groupings can be derived from the musical structure of a sequence. The CLaMP model [238], which is based on the ABC notation that includes pipe characters to represent bars, considers a bar-based grouping. However, such musically informed groupings are 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 [161] where a vocabulary of subwords is statistically learned on a training corpus. These include **Byte-Pair Encoding (BPE)** [68, 201], WordPiece [199], or UnigramLM [122]. Some of them have been adapted for music to create musical subwords as tokens. The BPE algorithm is adapted for orchestral data [144] 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 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 [64], experiments studying the impact of BPE in music analysis tasks do not show a significant increase in performance [252], unlike BPE applied to text [201]. In a more restricted musical context, Mel2Word [173] implements BPE with monophonic tunes and enables the retrieval of style-specific motifs. Finally, UnigramLM subword tokenization is also specifically evaluated on music generation, applied to score-based music and guitar tablatures [123]. 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 a Sequence of Combinations of Musical Features. While sequences of elementary tokens need 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 its musical features into a single *super-token*. The choice of the type of super-tokens, the musical features encapsulated within them, and the method used to construct the vector representing each super-token are the key 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. [266] is based on the concatenation of multiple one-hot vectors describing pitch, duration, chords, and bar. Octuple [251] is instead based on the embedding of eight musical features which are concatenated to form the single vector representing a single note. Such homogeneous representations are also used by PiRhDy [138] encoding pitch classes and octave instead of MIDI value, and MMT [52] for multi-track music. Instead of vectors, MuseBERT [228] embeds matrices derived from a set of onset, pitch, and duration to describe both musical attributes with their relations. Beyond notes, the Chordinator model [40] encodes chords described by a root, a nature, extensions, and a set of notes composing the chord.

On the other hand, methods separating events by families have been developed to highlight the distinction between note events and structural events such as the beginning of a bar. For polyphonic music, MuMIDI [189] represents a token as a sum of the embeddings of bars, position, and tempo, with possibly note characteristics. Similarly, Compound Word [94] 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 [154]. This representation is also enhanced by Di et al. [48] in the context of video-to-music, by incorporating a token family related to rhythm, encapsulating rhythm density and strength. Unsupervised

Compound Word [214] is based on the original CPWord tokenization and includes BPE, which learns the atomic element groupings instead of relying on pre-defined families resulting in variable-length composite tokens. REMI_Track [151] improves REMI+ and also combines composite tokens and BPE. Tokens are defined as 5-long vectors, with note-related elements of a token (pitch, velocity, duration) being possibly grouped together under a BPE-element, which has shown to improve inference efficiency when generating long sequences.

2.2 Comparing Tokenization Strategies

Various tokenization methods naturally lead to various performance depending on tasks or data. 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 [49]. 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 [135]. Fradet et al. [65] 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 analysis tasks [252].

The MidiTok Python package [63] 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 [252] or a component integrated into a processing pipeline coupled with the HuggingFace library [123]. Similarly, Musicaiz [88] offers a tokenization framework, with extensive visualization, generation, and analysis frameworks for symbolic music.

2.3 Preparing Music Data for Model Processing

The previous sections describe music encoded as sequential elements and operations that can be applied to them while keeping their high-level musical meaning. When used as inputs of most machine learning models, these elements need to be *embedded* or converted into numerical values so that the model can process them. Text, subwords, words, or documents need to be projected into a particular space to be processed [137], 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*, or BoW [114, Chap. 4]. This representation is improved by TF-IDF (Term Frequency–Inverse Document Frequency) [114, 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 [233], mode classification in Gregorian chant [35], or Chinese folk music clustering [254]. 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 representing words as vectors in a dense and continuous 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 [162] 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 [98, 152]. Such chord embeddings exhibit musical relations and are evaluated on downstream tasks like chord prediction and composer

classification [126]. PitchClass2Vec [127] 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 [92] uses Word2Vec on monophonic melodies by assuming such words as musical phrases segmented by GTTM rules [130]. Word2Vec has also been adapted for polyphonic music [89], 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 [29].

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 (i.e., where chords are identified by their function relative to an overall tonality). Technically, contextual embeddings are built concurrently with model training, such as recurrent or attention-based models described in Section 3. Yet, while analyses of learned contextual embeddings are numerous in NLP [145], 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 a **Long Short-Term Memory (LSTM)** model [69] or from BERT (Bidirectional Encoder Representations from Transformers) embeddings [85]. Fradet et al. [64] have shown that the learned contextual embedding space from BERT is more structured than the one learned from GPT-2. 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 [138] encode such musical-specific context encapsulating melodic and harmonic contexts.

3 NLP Models for Symbolic Music Processing

This section reviews *models* that have been borrowed or inspired from NLP and adapted to address MIR tasks. This transfer primarily arises from the temporal nature of music, which facilitates its representation as sequences of elements, as presented in Section 2, thus allowing its processing by NLP-based models, which are mostly data driven. Historically, shallow machine learning models were prominent for many years in NLP. Starting in the 1990s, in particular, models based on recurrent cells, like RNNs, became widely popular. This trend continued until the breakthrough of attention-based models in the mid-2010s. MIR studies also followed these trends, adapting these models to symbolic music in various ways.

3.1 Corpora for Data-Driven Models

Although NLP and symbolic MIR research include a number of rule-based approaches, most state-of-the-art models today are data driven. For this purpose, multiple datasets have been compiled, particularly through common crawl in NLP [231], and have been released for model training. In symbolic music, multiple collections of MIDI files have been compiled for generative models training. These include large crawled MIDI collections such as LakhMIDI [185] or MetaMIDI Dataset [59], and specific music genres or instrumentations such as orchestral music [144], piano music [86], chorales [10], folk tunes [198], or pop music [227]. Other datasets with specific music representations, such as guitar tablatures [195] or chords-only [42], have been built for non-MIDI generative systems. Datasets linking symbolic music and other types of data are built for multimodal models for audio-MIDI alignment [86] or, more recently, text-to-MIDI [159] and

video-to-MIDI [16]. For an in-depth overview of music generation datasets, refer to the work of Ji et al. [107].

Beyond music generation, a number of symbolic music datasets have also been released for traditional MIR tasks. While non-annotated datasets can automatically be compiled, such as through web crawling, music analysis datasets typically require annotations by experts [46]. Most of the datasets presented previously also include genre or composer annotations for full sequence classification tasks. Similarly, the EMOPIA dataset [102] includes annotations of valence and arousal for emotion classification. With more local annotations, the TAVERN dataset [46] includes chord and phrase annotations of a corpus of classical themes and variations. The *When in Rome* dataset [73] specifically gathers annotations of chord functions of a larger music era range, for automatic functional harmony analysis.

3.2 Shallow Models

Prior to the widespread adoption of data-driven methods, natural language modeling was mostly addressed by rule-based systems, such as formal grammars. A formal grammar is a set of rules that defines the syntactic structure of sentences in a language, specifying how words and phrases can be combined to form grammatically correct sentences. They are used in text for syntactic parsing or semantic analysis such as dependency parsing, representing text as tree structures. Musical grammars [190] have also been formalized, particularly based on harmony, for tasks such as jazz chord analysis [208]. Generative grammars [26], aiming at generating sentences based on rules, have also been applied in music. For instance, the "Generative Theory of Tonal Music" [130] is based on musical harmony and tension rules to generate music.

Such grammars are often used in conjunction with shallow sequential models. **Hidden Markov Models (HMMs)** and **Conditional Random Fields (CRFs)** are sequential models that were applied to NLP tasks much earlier than symbolic music. HMMs rely on the assumption that each observed element of a sequence is the result of a hidden process with the Markov property (short span dependencies). As a generalization of HMMs, CRFs are discriminative models that can impose dependencies on arbitrary elements of the sequence. In NLP, HMMs and CRFs have been implemented for part-of-speech tagging [125], named entity recognition [157], or text classification [66]. These models have then been widely used in early MIR studies for various symbolic music tasks such as style classification [220], melody prediction [203], harmonization [77], generation [218], chord recognition [156], or key detection [165].

Neural networks have since demonstrated greater performances, leading to architectures such as **Recurrent Neural Networks (RNNs)** that offer an alternative way of representing time and therefore handling sequential data.

3.3 Recurrent Models

RNNs [193] are a class of artificial neural networks designed to process sequential data by maintaining a hidden state that captures information about previous inputs. They have multiple applications in NLP, as well as in other fields that involve sequential dependencies, such as time series prediction. In MIR, only a few studies used raw RNN models, such as RNN-RBM [10] or RNN-DBN [71] combining RNN, Restricted Boltzmann Machine, and Deep Belief Network for polyphonic music generation. Such RNNs, however, have been shown to suffer from the issue of vanishing gradient occurring with long sequences, which is often the case in symbolic music.

LSTM [93] has been developed to address this issue and has since been widely adopted in multiple domains. An other improvement of recurrent networks then emerged with the introduction of **Gated Recurrent Units (GRUs)** [23]. Compared to LSTM models, GRUs are based on a simpler architecture, thereby reducing the total number of parameters and consequently reducing training

time, while maintaining similar performances to LSTM [30]. Multiple studies have implemented these models for harmonic analysis [19], music infilling [82], chorale harmonization [83], orchestral music generation [149], or expressive performance generation [170].

These recurrent layers are often part of larger architectures and might be improved through various mechanisms. They can be part of **Generative Adversarial Networks (GANs)** [72], which are models consisting of a generator and a discriminator, trained simultaneously through adversarial training to generate realistic data. Such architecture has been developed for chord-conditioned generation [216] or lyrics-conditioned melody generation [248]. Recurrent layers can also be used in **Variational Auto-Encoders (VAEs)** [120], which are generative models that learn to encode and decode data in a probabilistic way, allowing for the generation of new samples while capturing the underlying structure of the input data. While being mainly implemented for generative purposes [14, 191], the learned latent space of VAEs can also be analyzed, revealing particular directions representing musical aspects such as speed or repetitiveness [217] in the same way as text VAEs can highlight semantic relations [45]. Specific architectures based on LSTM or GRU have also been improved with mechanisms such as *attention* [4], which aims at giving different weights of importance to the elements of the processed sequence. This mechanism can be used with symbolic music for enhancing overall coherence in a multi-track arrangement task [263] or enforcing temporal structure [106]. This mechanism is still used in recent LSTM-based models [84]. Finally, recurrent layers have also been employed in models trained on symbolic MIR tasks using other paradigms, particularly reinforcement learning, based on a model trained to make decisions by interacting with an environment by rewarding or penalizing it. The choice of these rewards is often based on musical rules, such as pitch entropy or chords [124] or note intervals and repetitiveness [111]. An exhaustive overview of recurrent models used for symbolic MIR tasks is available on the companion web page of this survey.[10] From the end of the 2010s and the breakthrough of Transformer models [219], several state-of-the-art models have been derived from this model.

3.4 Attention-Based Models

Attention is a mechanism proposed by Bahdanau et al. [4], initially as an improvement of RNNs (Section 3.3). Vaswani et al. [219] then introduced *Transformers*, showing that a model based solely on attention—without using any recurrent mechanism—can outperform state-of-the-art results in NLP. More precisely, Transformers are based on a *self-attention* mechanism and *multi-head attention* blocks. They offer two main improvements to RNNs. 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 LSTMs. Whereby during the weight update process of the recurrent network, known as back propagation through time, such recurrent models often struggle in capturing long-term dependencies between words [168]. This phenomenon is also true for music generation [90].

Transformers have been applied to symbolic music representations, but also in a variety of other domains, such as computer vision [54] or audio [53]. Their use has been greatly facilitated with the development of libraries, such as AllenNLP [70], FairSeq [171], or, more predominantly, HuggingFace [232]. This last library offers model architectures, pre-trained models, tokenizers, and various utilities to simplify the development and deployment of NLP applications. As a result, numerous MIR studies have started utilizing HuggingFace by leveraging its tools and resources for musical tasks. These include implementations of subword tokenizers (Section 2.1.2) such as

[10] https://github.com/dinhviettoanle/survey-music-nlp#recurrent-models

BPE [201] or Unigram [122] used by Kumar and Sarmento [123] and model implementations such as BERT [47] for MidiBERT [28] or GPT-2 [184] used in MMM [58].

In this section, we propose an overview of attention-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 3.4.1). 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 3.4.2). The model architecture, based on Transformer encoders, decoders, or combining different types of data, influences how music is processed. Finally, we present the enhancements of the Transformers' internal mechanism to specifically process symbolic music data (Section 3.4.3). Summaries of these Transformer-based models for symbolic MIR are presented in Tables 3 and 4.

3.4.1 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 step on a generic task followed by a fine-tuning step on one or multiple tasks. This approach is at the heart of LLMs in NLP. From a musical point of view, pre-trained models aim first at modeling or *understanding* music globally, similarly to the understanding of natural language in NLP [256], from which specific downstream tasks can then be derived via fine-tuning.

3.4.1.1 End-to-end models. End-to-end models are trained for a specific task. They include Transformer-based GANs [72], resulting in models for free generation [164] or emotion-driven generation [167]. Other systems rely on Transformer-based VAEs [120] for priming-conditioned generation [110], chord-conditioned generation [25], lyrics-conditioned generation [57], or artistic-controllable generation [222]. This last task is also performed in a multi-track context [128], with fine-grained control of the musical features at the track level.

End-to-end models also include several data-specific models designed to process musical data beyond notes. The Chordinator [40] model handles chord data and is based on a minGPT architecture,[11] without its pre-training process. Several models are trained on guitar tablatures, for tablature generation [22], metadata-conditioned generation [195], style-driven generation [197], or instrument-conditioned generation for bands [196]. 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, 21] or style-annotated datasets [3].

3.4.1.2 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 NLP-derived pre-trained models have switched to Transformer-based architectures both in NLP and MIR.

State-of-the-art pre-trained language models include BERT [47]. BERT is based on a bidirectional training approach and 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

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

Table 3. *End-to-End* Transformer-Based Models Applied to Symbolic Music[12]: Such Models Are Directly Trained on Specific Tasks

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Encoder-only architecture							
MTBert [258]	(2023)	BERT (no pre-training)	–	4-part chorales	Interval + duration (event based)	Fugue form analysis	✗
Decoder-only architecture							
Music Transformer [99]	(2018)	Tf. decoder	Relative attention	Piano/Choral	MIDI-like	Priming/Harmonization	✓
Pop Music Transformer [101]	(2020)	Transformer-XL	–	Piano	REMI	Free generation/Priming	✓
Jazz Transformer [239]	(2020)	Transformer-XL	–	Lead sheet	REMI derived (Chords)	Free generation	✓
PopMAG [189]	(2020)	Transformer-XL	–	Multi-track	MuMIDI	Accompaniment generation	✗
Di et al. [48]	(2021)	Tf. decoder	–	Multi-track	CPWord derived (Rhythm family)	Video-to-music	✓
Chang et al. [18]	(2021)	XLNet	Relative bar encoding	Piano	Compound Word	Infilling	✓
Compound Word Tf. [94]	(2021)	Linear Tf. decoder	–	Piano	Compound Word	Free generation/Priming	✓
Sarmento et al. [195]	(2021)	Transformer-XL	–	Guitar tabs + multi-track	DadaGP	Metadata-conditioned gen.	✓
ComMU [128]	(2022)	Transformer-XL	–	Multi-track	REMI + metadata	Metadata-conditioned gen. Multi-track combination	✓
SymphonyNet [144]	(2022)	Linear Tf.	3-D positional encoding	Orchestral	MMR	Free generation/Priming Chord-conditioned generation	✓
Multitrack Music Tf. [52]	(2023)	Tf. decoder	–	Orchestral	MMT	Free generation/Priming Instr.-conditioned generation	✓
GTR-CTRL [196]	(2023)	Transformer-XL	–	Guitar tabs + multi-track	DadaGP	Instr.-conditioned generation Genre-conditioned generation	✗
Choir Transformer [261]	(2023)	Tf. decoder	Relative attention	4-part chorales	Chord + pitch (event based)	Harmonization	✓
Guo et al. [80]	(2023)	Tf. encoder w/ custom attention	Fundamental music embedding RIPO attention	Monophonic	FME	Priming	✓

(Continued)

[12]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 3. Continued

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Compose & Embellish [240]	(2023)	Tf. decoder	–	Piano	REMI	Lead sheet priming Accompaniment refinement	✓
RHEPP-Transformer [211]	(2023)	Tf. decoder	–	Piano	Octuple	Expressive performance gen.	✓
Chordinator [40]	(2024)	minGPT (no pre-training)	–	Chords	Custom chord features (+ MIDI array)	Chord generation	✓
Encoder-decoder architecture							
Transformer-VAE [110]	(2020)	Tf. encoder-decoder	–	Monophonic	Pitch + duration (time-slice based)	Priming	✗
Harmony Transformer [21]	(2021)	Tf. encoder-decoder	–	Piano	Piano roll time-slices	Roman Numeral Analysis	✓
Makris et al. [153]	(2021)	Tf. encoder-decoder	–	Lead sheet	Enc.: bar features Dec.: chord + pitch + dur.	Emotion-conditioned gen.	✓
Liutkus et al. [147]	(2021)	Performer	Stochastic positional encoding	Multi-track	REMI/MIDI-like derived (multi-track)	Free generation Groove continuation	✓
Gover and Zewi [75]	(2022)	BART	–	Piano	REMI derived (hands tokens)	Arrangement generation	✗
Museformer [247]	(2022)	Tf. encoder-decoder w/ custom attention	Fine-/coarse-grained attention Bar selection	Multi-track	REMI	Free generation	✓
Theme Transformer [205]	(2022)	Tf. encoder-decoder	Theme-aligned pos. enc.	Multi-track	REMI derived (theme tokens)	Theme-conditioned generation	✓
FIGARO [222]	(2022)	Tf. encoder-decoder	–	Multi-track	REMI+	Controllable generation	✓
MuseMorphose [241]	(2023)	Tf. enc + Transformer-XL	In-attention conditioning	Piano	REMI derived (multi-track)	Style transfer Controllable generation	✓
Accomontage 3 [259]	(2023)	Tf. encoder-decoder	Instrument embedding	Multi-track	Piano roll time-slices	Accompaniment generation	✓
TeleMelody [113]	(2022)	Tf. encoder-decoder	–	Monophonic	Bar + position + pitch + duration	Lyrics-to-melody	✓
MuseCoco [150]	(2023)	Text2Attr.: BERT Attr2Music: Linear Tf.	–	Multi-track	REMI	Text-to-MIDI	✓
MelodyT5 [237]	(2024)	T5	–	Monophonic	ABC notation	Melody gen./harmonization Melody segmentation	✓

(Continued)

Table 3. Continued

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Encoder-decoder architecture							
Composer's Assistant 2 [155]	(2024)	T5	–	Multi-track	REMI+ derived (text format)	Infilling Controllable generation	✗
BandControlNet [151]	(2024)	Tf. encoder-decoder	Structure enhanced self-attention	Multi-track	REMI_Track	Controllable generation	✓
Model combinations							
Transformer-GAN [164]	(2021)	Generator: Tf.-XL Discriminator: BERT	–	Piano	MIDI-like	Free generation	✓
Choi et al. [25]	(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 [180]	(2022)	Bi-LSTM - Tf. decoder	–	Lead sheet	Bar + position + melody + chord (time-slice based)	Free generation	✗
Makris et al. [154]	(2022)	Bi-LSTM - Tf. decoder	–	Multi-track	CPWord derived	Drums accomp. generation	✓
Neves et al. [167]	(2022)	Generator: Linear Tf. Discriminator: Linear Tf.	Local prediction map	Piano	REMI	Emotion-conditioned gen.	✓
Q&A [260]	(2023)	PianoTree-VAE Tf. decoder	Instrument embedding	Multi-track	Piano roll time-slices	Accompaniment generation	✓
Duan et al. [57]	(2023)	Generator: Tf. encoder Discriminator: LSTM	–	Monophonic	Pitch + duration + rest (event based)	Lyrics-to-melody	✗
Video2Music [115]	(2023)	GRU + Tf. encoder-decoder	–	Multi-track	MIDI-like	Video-to-music	✓

Models are grouped by architecture. Details indicated in the *Representation* column depict the specific adaptations brought to an initial tokenization strategy. The last column indicates the code availability.

Table 4. *Pre-Trained* Transformer-Based Models Applied to Symbolic Music[13]: 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 [228]	(2021)	BERT	Generalized relative pos. enc.	Piano	MuseBERT repr.	Controllable generation/Chord analysis Accompaniment refinement	✓
MidiBERT-Piano [28]	(2021)	BERT	–	Piano	REMI Compound Word	Melody extraction/Velocity prediction Composer /Emotion classification	✓
MusicBERT [251]	(2021)	RoBERTa	Bar-level masking	Multi-track*	Octuple	Melody completion/Accomp. suggestion Genre /Style classification	✓
DBTMPE [181]	(2021)	Tf. encoder	–	Multi-track	Pitch combinations + duration (event based)	Style classification	✗
MRBERT [134]	(2023)	BERT	Melody/rhythm cross attention	Lead sheet	Pitch + duration (event based)	Free generation/Infilling Chord analysis	✗
SoloGPBERT [197]	(2023)	BERT	–	Guitar tabs	DadaGP	Guitar player classification	✗
Shen et al. [204]	(2023)	MidiBERT-Piano	Quad-attribute masking + Key prediction pre-training tasks	Multi-track	CPWord simplified	*Same as MidiBERT-Piano* [28]	✗
CLaMP [238]	(2023)	Text enc.: DistilRoBERTa Music enc.: BERT	–	Lead sheet	ABC notation derived	Text-based semantic music search Music recommendation/classif.	✓
Decoder-only architecture							
LakhNES [51]	(2019)	Transformer-XL	–	Multi-track	MIDI-like	Free generation	✓
Musenet [174]	(2019)	GPT-2	Timing embedding Structural embedding	Multi-track*	MIDI-like	Priming	✗
MMM [58]	(2020)	GPT-2	–	Multi-track	MultiTrack repr.	Free generation/Inpainting Priming /Controllable generation	✓
Zhang and Callison-Burch [253]	(2023)	GPT-3	–	Drums	Drumroll time-slices	Priming	✓
ChatMusician [249]	(2024)	Llama-2	–	Monophonic	ABC Notation	Text-to-ABC	✓
ComposerX [44]	(2024)	GPT-4	–	Monophonic	ABC notation	Text-to-ABC	✓
MuseBarControl [206]	(2024)	Linear Tf.	Auxiliary task pre-adaptation	Piano	REMI	Controllable-/Chord-conditioned gen.	✗

(Continued)

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

Table 4. Continued

Model		Base model	MIR mechanism	Data	Representation	Tasks	Code
Encoder-decoder architecture							
MusIAC [79]	(2022)	Tf. encoder-decoder	–	Multi-track	REMI	Infilling/Controllable generation	✓
Fu et al. [67]	(2023)	MusicBERT + Music Tf.	–	Multi-track	Octuple	*Same as MusicBERT* [251]	✗
Multi-MMLG [257]	(2023)	XLNet + MuseBERT	–	Multi-track	CPWord derived	Melody extraction	✗
PianoBART [139]	(2024)	BART	Multi-level object masking	Piano	Octuple	Priming + *same as MidiBERT-Piano* [28]	✓
Comparative studies							
Ferreira et al. [61]	(2023)	GRU, Performance-RNN GPT-2, Music Tf., MuseNet (Tf. decoders)	–	Piano	MIDI-like	Free generation	✓
Wu and Sun [236]	(2023)	BERT (Tf. encoder) GPT-2 (Tf. decoder) BART (Tf. enc.-dec.)	–	Lead sheet	ABC notation	Text-to-ABC	✓

symbolic music have been proposed. MuseBERT [228] develops a specific representation merging musical attributes and relations and processed by the attention mechanism. MusicBERT [251] is a model designed based on RoBERTa [146] 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 than a MusicBERT only [67]. Instrument-specific BERTs have been implemented such as SoloGPBERT [197] for guitar tablatures, MRBERT [134] for lead sheets, or MidiBERT-Piano [28] for piano. This model is then extended beyond piano music and improved with musically meaningful pre-training tasks [204]. BART [131] is also a model pre-trained via token masking and is used by PianoBART [139], implementing multiple-level token masking and resulting in better performance than other BERT models.

GPT (Generative Pre-trained Transformer) [183] 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 [184], GPT-3 [13], and GPT-4 [15]. For symbolic music, Musenet [174] and MMM [58] are based on GPT-2 and are trained for conditioned generation. Another approach has been implemented for drum music generation [253]: music is represented as textual data, 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 [51] and DBTMPE [181] 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.

3.4.2 Model Architecture: Transformer Encoder/Decoder and Multimodal Models. Attention-based models can also be categorized by their architecture. In NLP, the first Transformer model for translation [219] was based on an encoder-decoder architecture. Since then, several NLP models based on either encoders [47], decoders [183], or with modified mechanisms have been proposed. MIR studies have 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, particularly for tasks like text-to-symbolic music. These multimodal models have found application in domains such as audio processing with MusicLM [1] or non-music fields such as image processing with Dall-E [187].

3.4.2.1 Encoder only. Encoders are based on a self-attention mechanism, allowing the learning of knowledge on the complete sequence. Bidirectional models, which are based on this encoder-only architecture, have led to symbolic music adaptations of BERT such as MuseBERT [228], MusicBERT [251], MidiBERT-Piano [28], MRBERT [134], and SoloGPBERT [197]. Going further, Han et al. [85] 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 [258] 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 in the following. Such Transformer encoders are also widely used as the discriminator module in GAN-based models [164, 255], initially developed for generation purposes. They are usually implemented followed by an encoder-decoder or decoder-only as the GAN generator.

3.4.2.2 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 [99] is based on a decoder-only model for priming and harmonization tasks, and is then reused by Sulun et al. [210] for emotion-conditioned generation. Generation is tackled by the MultiTrack Music Transformer [52] for instrument-conditioned generation then improved for genre control [243], the Choir Transformer [261] for four-part harmonization, Compose & Embellish [240] for lead sheet and piano accompaniment generation, and by Tang et al. [211] for expressive performance reconstruction. Decoder-only models can also be trained through a pre-training/fine-tuning process, particularly with GPT-based models, such as Musenet [174] or MMM [58]. By comparing multiple decoder-only architectures, such pre-trained decoder-only models appear to perform better in piano generation [61].

Several models combine recurrent models with Transformer decoders. Q&A [260] combines GRU-based PianoTree-VAEs with a Transformer decoder for arrangement generation. In the same way, Choi et al. [25] 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 [180] for long-term structure generation, where the LSTM captures note-level dependencies and Transformer decoders capture bar-level relations.

A limiting issue with Transformers is the quadratic complexity of the attention mechanism with respect to the sequence length, which induces long training times. The Linear Transformer [116] improves the attention mechanism with a linear complexity. The Compound Word Transformer [94] takes advantage of this computational optimization, coupled with its shorter sequence representation, for piano music generation. SymphonyNet [144] 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 [39], 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 [128], piano music [101, 164, 241], lead sheets [136, 239], or guitar tablatures [22, 195–197]. Chang et al. [18] implement an improved Transformer-XL, XL-Net [245], a Transformer-based model that can attend to past and future in the same way as BERT, while maintaining an autoregressive predicting order.

3.4.2.3 Encoder-decoder. Finally, following the architecture of the vanilla Transformer, multiple models for symbolic MIR implement an encoder-decoder architecture. Functional harmony analysis has been tackled by the Harmony Transformer [20, 21]. The model implements this architecture, where the encoder has a chord segmentation role and the decoder infers the chord symbol.

For generative purposes, such architectures are used with an encoder that analyzes musical constraints and a decoder that generates musical content. Makris et al. [153] implement similar architectures, with an encoder analyzing chord valence that conditions an auto-regressive decoder for a generation task. In the Theme Transformer model [205], 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 [79] 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 [257] 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. T5 [186] is an encoder-decoder model developed in NLP to handle text-to-text tasks. The model has been adapted for music by MelodyT5 [237] for melody-related tasks or Composer's Assistant [155] for an infilling task, both using textual representations of music to leverage the text-to-text characteristics of the backbone model. In NLP, encoder-decoder models are often implemented for translation

purposes [219]. Gover and Zewi [75] implement BART [131], 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 [259] for multi-track music with an encoder/multiple decoders architecture. This multiple independent decoder architecture is also implemented in BandControlNet [151] and is called *cross-track Transformer*, which is shown to improve fidelity-related metrics in a controllable generation task. Finally, this encoder-decoder architecture is largely used in autoencoder architectures. The Transformer VAE [110] implements a sampling step from a latent space, from which keys and values are derived for the cross-attention mechanism. MuseMorphose [241] and FIGARO [222] are models based on VAEs, developed for controllable symbolic music generation, which use their latent space representations as constraints.

3.4.2.4 Multimodal models. A variety of MIR systems have been developed to integrate other types of data such as text or video, in combination with symbolic music. In symbolic MIR, studies have explored models linking text and music, including a task of lyric-to-melody with TeleMelody [113] processing musical high-level features or operating at the syllable level [57]. Text-to-image systems have been gaining in popularity these past few years, resulting naturally in text-to-music systems in both audio [1] and symbolic music. MuseCoco [150] performs this text-to-MIDI task. However, most text-to-symbolic-music tasks currently process ABC notation, as this encoding is already in a textual format [236]. ChatMusician [249] is based on Llama-2 [215] and is framed as a music chatbot that can write ABC notation music and chat with a user about music theory knowledge. GPT-4 is able to perform such a text-to-ABC task, among multiple other tasks [15], but struggles at modeling musical concepts such as harmony. To overcome this issue, this task is split into multiple musically meaningful subtasks in ComposerX [44], which uses GPT-4 for melody generation, harmonization, and instrument selection. Finally, beyond generative tasks, CLaMP [238] integrates two BERT-based models—one for text encoding and the other for music encoding—for 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. [48] generate music for videos that are analyzed in terms of motion speed and saliency conditioning the generated music rhythm. Kang et al. [115] add a semantic and emotion analysis of the scene, and more specifically generate chords matching these video features.

3.4.3 Adapting Attention Models' Inner Mechanisms to Symbolic 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 in this different context.

Transformers implement a self-attention mechanism, which can be easily interpreted by visualizing it. Such visualization can show differences between attention heads being more or less specialized in chords or melody [95]. 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 [148], musical phrases or modulating sequences [109], or consonant musical intervals [52].

Multiple MIR studies have also developed positional encodings and customized for the specificities of music. With the Music Transformer model [99], a *relative positional self-attention* mechanism is developed for music generation enabling the processing of much longer sequences. Similarly, *stochastic positional encoding* [147] 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* [144] in which the track order is permutation invariant, unlike notes or bars that must remain time dependent. Musically meaningful positional encodings have been developed based

on notes attributes and relations [228], bars [18], musical themes [205], structure and musical time [174], or instruments [259, 260].

The attention mechanism itself has also been adapted for symbolic music. The Museformer model [247] 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. *RIPO (Relative Index, Pitch and Onset) attention* [80] is proposed with *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 [241] includes an *in-attention conditioning* that takes into account constraints in the self-attention computation. *Structure-enhanced self-attention* from BandControlNet [151] incorporates a similarity score between bars in the attention computation to enhance the structure consistency of tracks. For lead sheet data, a melody/rhythm cross attention is implemented in MRBERT [134], 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 [72], a *local prediction map* [167] is proposed so that the discriminator also specifies which parts of the generated sequence is real or generated. Pre-trained models, particularly masked language models, are usually pre-trained on a token prediction task from a masked sequence and a next sentence prediction task [47]. For symbolic music, MusicBERT [251] 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* [204]. Going further, PianoBART [139], which also uses an Octuple representation, implements a multi-level object masking strategy, where the masked token can be at the level of an Octuple-element, the whole Octuple, or ranging over multiple bars. 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. [204] also propose an analogous pre-training task with next sentence prediction with *key prediction*. MuseBarControl [206], a Linear Transformer for controllable generation, implements a pre-training task aiming at directly incorporating control signals during the pre-training step to improve the resulting bar-level controllability.

4 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, given that both modalities still share several differences as discussed in Section 1.1. We believe that incorporating such reflections as well as common practices from the NLP field could help guide future directions in the MIR field.

Data Availability. Text data differ from symbolic music data by a much wider availability. For example, LLMs such as GPT-3 [13] are trained on datasets containing 300B tokens. Compared to symbolic music, multiple models [58, 222] are trained on the LakhMIDI dataset, which is composed of 175k songs, resulting in only 26M tokens using a basic MIDI-like tokenization. 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. However, one way to expand symbolic music datasets could be through the use of audio datasets transcribed into symbolic music data. Audio-to-symbolic transcription tools have

shown strong performance [104] and could be leveraged to significantly increase the volume of symbolic music data.

In addition to the limited quantity of music data, the diversity of the available datasets may also be restricted. Similarly to classical music data which is largely biased toward Western music [86], contemporary music such as pop music may be stemming from non-western countries [227] but is still restricted to tonal music. In generative tasks, these biases in training data are naturally reflected in the generated content.

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 represented by 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 [235]. Such models have been developed for Chinese NLP and take these radicals into account [212].

Toward Lighter Models. In the field of NLP, various studies have focused on developing computationally efficient yet lighter models [264], especially with the rise of LLMs. 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 [74]. This distillation process implements a lightweight student network that is trained to reproduce a pre-trained teacher network. In NLP, this has led to lightweight models such as DistilBERT [194]. 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 [160] and can help model explainability [221]. Finally, model design optimizations for lightweight processing have been developed, such as token skipping in PoWER-BERT [76] or sliding window attention with cache in Mistral 7B [108]. In MIR, such advances toward lighter models have been tackled for audio music [55].

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]. However, recent models are increasingly requiring higher computational power, such as the use of 4×40 GB GPUs [206]. Therefore, there is a growing recognition of the efficacy of lighter models for symbolic music data, including the development of Compound Words [94] for smaller sequences, or smaller vocabulary resulting in smaller embeddings [135]. 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 music generation, including improvisation where an instantaneous inference time is required.

Toward 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 [256]. 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 [33], encompassing syntactic or semantic information retrieval [119]. 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 [225]. In the context of Transformers,

beyond embeddings, multiple representations can be analyzed [11], particularly 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 [117, 158]. 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 [121]. LLMs developed for chat can also be evaluated in their reasoning [249, 262], assessing their musical understanding and knowledge for future human-computer co-creation systems. Going further, providing interpretable tools that align with human behavior 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 [38]. 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. Only a few studies have considered linking a model's behavior with musicological aspects such as cadences [148] or chord progressions [37]. 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.

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 [224], to evaluate language understanding. Other specific NLP benchmarks have also been developed, such as cross-lingual benchmarks [140] or domain-specific benchmarks [176].

In symbolic MIR, there is currently an apparent lack of standardized benchmarks. Bundling of datasets, tasks, and evaluation metrics for symbolic music data may provide frameworks to compare and evaluate models. The re-introduction of MIREX challenges[14] in 2024 is an encouraging step toward model benchmarking. However, such challenges have mainly covered audio tasks. With the recent spread of text LLMs capable of processing ABC notation, ZIQI-Eval [132] has been proposed to objectively compare models trained to answer multiple choice music-related questionnaires. The 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 can be even more subjective [246], particularly when performed by non-experts [2]. However, MIREX's symbolic music generation tasks still rely on such listening tests for evaluation. In addition, objective evaluation metrics have been proposed [123, 239]. Valuable contributions regarding these benchmarking issues can be an evaluation toolkit library aiming at retrieving objective 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. A key challenge in music generation is that each model is typically specialized in a specific task.

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 [5] aims to represent long documents by implementing linear complexity

[14] https://www.music-ir.org/mirex

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 symbolic music tasks are more similar to textual character-level tasks. On the representation side, BERT-sentence [188] 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 segmentation is a recurrent issue, such textual sentence-derived representation holds potential relevance. In more practical cases, pattern matching is often used in incipit search engines such as RISM:[15] an embedding-based query method can improve the tool's flexibility.

Finally, beyond NLP and the excitement of the general public for tools based on natural language generation, another trend stemming from research studies is image generation, particularly text-to-image. Image processing models have already been used for symbolic music, including convolutional neural networks [24], and the recent rise of *diffusion models* in this field has motivated its adaption for music. Numerous recent models integrate state-of-the-art techniques from NLP and image processing, using diffusion models coupled with Transformer blocks for music generation [133, 163], also leading to tutorials on diffusion models for music at ISMIR 2024.[16] Therefore, as observed in recent publications and preprints (see Figure 3), a current trend from recent MIR studies is to adapt such diffusion 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 symbolic MIR frequently draws inspiration from methods employed in NLP. Musical adaptations of NLP tools are organized in this survey following 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 multidimensionality 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. Deep learning models were historically based on recurrent models until the breakthrough of Transformers in NLP, which then spread the development of several attention-based models for symbolic MIR. However, acknowledging the particular characteristics of music in comparison with text, many models have incorporated music-specific mechanisms into Transformers, such as positional encoding or attention mechanisms.

Despite the promising 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

[15] https://opac.rism.info

[16] https://ismir2024.ismir.net/tutorials#page-section-2

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.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and feedback, as well as the Algomus and MAGNET teams for the fruitful discussions.

References

[1] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. 2023. MusicLM: Generating music from text. *arXiv:2301.11325* (2023).

[2] Teresa M. Amabile. 1982. Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology* 43, 5 (1982), 997–1013.

[3] Simone Angioni, Nathan Lincoln-DeCusatis, Andrea Ibba, and Diego Reforgiato Recupero. 2023. A Transformers-based approach for fine and coarse-grained classification and generation of MIDI songs and soundtracks. *PeerJ Computer Science* 9 (2023).

[4] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR'15)*.

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

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

[7] Mireille Besson and Daniele Schön. 2001. Comparison between language and music. *Annals of the New York Academy of Sciences* 930 (2001), 232–258.

[8] Rens Bod. 2002. A unified model of structural organization in language and music. *Journal of Artificial Intelligence Research* 17 (2002), 289–308.

[9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[10] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the International Conference on Machine Learning (ICML'12)*.

[11] Adrian M. P. Braşoveanu and Răzvan Andonie. 2020. Visualizing Transformers for NLP: A brief survey. In *Proceedings of the 2020 24th International Conference Information Visualisation (IV'20)*. 270–279.

[12] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. 2020. *Deep Learning Techniques for Music Generation*. Vol. 1. Springer.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS'20)*, Vol. 33. Curran Associates, Inc., 1877–1901.

[14] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. 2018. MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'18)*.

[15] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv:2303.12712* (2023).

[16] Igor Cardoso, Rubens O. Moraes, and Lucas N. Ferreira. 2024. The NES video-music database: A dataset of symbolic video game music paired with gameplay videos. In *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG'24)*. Association for Computing Machinery, New York, NY, USA, Article 19, 6 pages.

[17] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2021. Evaluation of text generation: A survey. *arXiv:2006.14799* (2021).

[18] Chin-Jui Chang, Chun-Yi Lee, and Yi-Hsuan Yang. 2021. Variable-length music score infilling via XLNet and musically specialized positional encoding. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21)*.

[19] Tsung-Ping Chen and Li Su. 2018. Functional harmony recognition of symbolic music data with multi-task recurrent neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'18)*. ISMIR, 90–97.

[20] Tsung-Ping Chen and Li Su. 2019. Harmony Transformer: Incorporating chord segmentation into harmony recognition. In *International Society for Music Information Retrieval Conference (ISMIR)*. ISMIR, 259–267.
[21] Tsung-Ping Chen and Li 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 (2021), 1–13.
[22] Yu-Hua Chen, Yu-Hsiang Huang, Wen-Yi Hsiao, and Yi-Hsuan Yang. 2020. Automatic composition of guitar tabs by Transformers and groove modeling. In *International Society for Music Information Retrieval Conference (ISMIR)*.
[23] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734.
[24] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2392–2396.
[25] Kyoyun Choi, Jonggwon Park, Wan Heo, Sungwook Jeon, and Jonghun Park. 2021. Chord conditioned melody generation with Transformer based decoders. *IEEE Access* 9 (2021), 42071–42080.
[26] Noam Chomsky. 1957. *Syntactic Structures*. De Gruyter Mouton, Berlin, New York.
[27] Noam Chomsky. 1980. *Human Language and Other Semiotic Systems*. Springer US, Boston, MA, 429–440.
[28] Yi-Hui Chou, I.-Chun Chen, Chin-Jui Chang, Joann Ching, and Yi-Hsuan Yang. 2021. MidiBERT-Piano: Large-scale pre-training for symbolic music understanding. *arXiv:2107.05223* (2021).
[29] Ching-Hua Chuan, Kat Agres, and Dorien Herremans. 2020. From context to concept: Exploring semantic relationships in music with Word2Vec. *Neural Computing and Applications* 32, 4 (2020), 1023–1036.
[30] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of the NIPS 2014 Workshop on Deep Learning*.
[31] Rudi Cilibrasi, Paul VitÁnyi, and Ronald de Wolf. 2004. Algorithmic clustering of music based on string compression. *Computer Music Journal* 28, 4 (12 2004), 49–67.
[32] Darrell Conklin and Ian H. Witten. 1995. Multiple viewpoint systems for music prediction. *Journal of New Music Research* 24, 1 (1995), 51–73.
[33] Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco 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*. 2126–2136.
[34] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and controllable music generation. In *Adv. Neural Inf. Process. Syst.*
[35] Bas Cornelissen, Willem H. Zuidema, and John Ashley Burgoyne. 2020. Mode classification and natural units in plainchant. In *International Society for Music Information Retrieval Conference (ISMIR'20)*. 869–875.
[36] Débora C. Corrêa and Francisco Ap. Rodrigues. 2016. A survey on symbolic data-based music genre classification. *Expert Systems and Applications* 60 (2016), 190–210.
[37] Nicole Cosme-Clifford, James Symons, Kavi Kapoor, and Christopher Wm. White. 2023. Musicological interpretability in generative Transformers. In *4th International Symposium on the Internet of Sounds*. 1–9.
[38] R. L. Crocker. 1966. *A History of Musical Style*. McGraw-Hill.
[39] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan 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*. 2978–2988.
[40] David Dalmazzo, Ken Déguernel, and Bob L. T. Sturm. 2024. The Chordinator: Modeling music harmony by implementing Transformer networks and token strategies. In *Artificial Intelligence in Music, Sound, Art and Design*. Lecture Notes in Computer Science, Vol. 14633. Springer, 52–66.
[41] Adyasha Dash and Kat R. Agres. 2023. AI-based affective music generation systems: A review of methods, and challenges. *arXiv:2301.06890* (2023).
[42] Jacopo de Berardinis, Albert Merono Penuela, Andrea Poltronieri, and Valentina Presutti. 2023. ChoCo: A chord corpus and a data transformation workflow for musical harmony knowledge graphs. *Scientific Data* 10, 1 (2023), 641.
[43] Douglas Dempster. 1998. Is there even a grammar of music ? *Musicae Scientiae* 2, 1 (1998), 55–65.
[44] Qixin Deng, Qikai Yang, Ruibin Yuan, Yipeng Huang, Yi Wang, Xubo Liu, Zeyue Tian, Jiahao Pan, Ge Zhang, Hanfeng Lin, et al. 2024. ComposerX: Multi-agent symbolic music composition with LLMs. *arXiv:2404.18081* (2024).
[45] Michel Deudon. 2018. Learning semantic similarity in a continuous space. *Adv. Neural Inf. Process. Syst. (NeurIPS)* 31 (2018), 994–1005.
[46] Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. 2015. Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'15)*.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[48] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng 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, New York, NY, USA, 2037–2045.

[49] Miguel Domingo, Mercedes García-Martínez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. 2023. How much does tokenization affect neural machine translation? In *Computational Linguistics and Intelligent Text Processing*. Lecture Notes in Computer Science, Vol. 13451. Springer, 545–554.

[50] Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. *arXiv:2005.05339* (2020).

[51] Chris Donahue, Huanru Henry Mao, Yiting Ethan Li, Garrison W. Cottrell, and Julian McAuley. 2019. LakhNES: Improving multi-instrumental music generation with cross-domain pre-training. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'19)*.

[52] Hao-Wen Dong, Ke Chen, Shlomo Dubnov, Julian McAuley, and Taylor Berg-Kirkpatrick. 2023. Multitrack Music Transformer. In *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'23)*. 1–5.

[53] Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*. 5884–5888.

[54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent. (ICLR)*.

[55] Constance Douwes, Giovanni Bindi, Antoine Caillon, Philippe Esling, and Jean-Pierre 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.

[56] Stephen Downie. 1999. *Evaluating a Simple Approach to Music Information Retrieval: Conceiving Melodic n-Grams as Text*. University of Illinois.

[57] Wei Duan, Yi Yu, Xulong Zhang, Suhua Tang, Wei Li, and Keizo Oyama. 2023. Melody generation from lyrics with local interpretability. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 3, Article 124 (2 2023), 21 pages.

[58] Jeff Ens and Philippe Pasquier. 2020. MMM: Exploring conditional multi-track music generation with the Transformer. *arXiv:2008.06048* (2020).

[59] Jeffrey Ens and Philippe Pasquier. 2021. Building the MetaMIDI dataset: Linking symbolic and audio musical data. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21)*. 182–188.

[60] Jose D. Fernández and Francisco Vico. 2013. AI methods in algorithmic composition: A comprehensive survey. *Artif. Intell. Res.* 48, 1 (2013), 513–582.

[61] Pedro Ferreira, Ricardo Limongi, and Luiz Paulo Fávero. 2023. Generating music with data: Application of deep learning models for symbolic music composition. *Applied Sciences* 13, 7 (2023).

[62] Johan Fornäs. 1997. Text and music revisited. *Theory, Culture & Society* 14, 3 (1997), 109–123.

[63] Nathan Fradet, Jean-Pierre Briot, Fabien Chhel, Amal El Fallah-Seghrouchni, and Nicolas Gutowski. 2021. MidiTok: A Python package for MIDI file tokenization. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21): Late-Breaking Demo Session*.

[64] Nathan Fradet, Nicolas Gutowski, Fabien Chhel, and Jean-Pierre Briot. 2023. Byte pair encoding for symbolic music. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2001–2020.

[65] Nathan Fradet, Nicolas Gutowski, Fabien Chhel, and Jean-Pierre Briot. 2023. Impact of time and note duration tokenizations on deep learning symbolic music modeling. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'23)*.

[66] Paolo Frasconi, Giovanni Soda, and Alessandro Vullo. 2002. Hidden Markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems* 18 (2002), 195–217.

[67] Yingfeng Fu, Yusuke Tanimura, and Hidemoto 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.

[68] Philip Gage. 1994. A new algorithm for data compression. *C~Users Journal* 12, 2 (1994), 23–38.

[69] Sebastian Garcia-Valencia. 2020. Embeddings as representation for symbolic music. *arXiv:2005.09406* (2020).

[70] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. *arXiv:1803.07640* (2018).

[71] Kratarth Goel, Raunaq Vohra, and J. K. Sahoo. 2014. Polyphonic music generation by modeling temporal dependencies using a RNN-DBN. In *Proceedings of the International Conference on Artificial Neural Networks and Machine Learning (ICANN'14)*. Springer International Publishing, Cham, 217–224.

[72] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NeurIPS'14)*. MIT Press, Cambridge, MA, USA, 2672–2680.

[73] Mark Gotham, Gianluca Micchi, Néstor Nápoles López, and Malcolm Sailor. 2023. When in Rome: A meta-corpus of functional harmony. *Transactions of the International Society for Music Information Retrieval* 6, 1 (Nov 2023), 150–166.

[74] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129 (2021), 1789–1819.

[75] Matan Gover and Oded Zewi. 2022. Music translation: Generating piano arrangements in different playing levels. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'22)*.

[76] Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In *International Conference on Machine Learning (ICML)*. PMLR, 3690–3699.

[77] Ryan Groves. 2013. Automatic harmonization using a hidden semi-Markov model. *AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*

[78] Yixing Guan, Jinyu Zhao, Yiqin Qiu, Zheng Zhang, and Gus Xia. 2018. Melodic phrase segmentation by deep neural networks. *arXiv:1811.05688* (2018).

[79] Rui Guo, Ivor Simpson, Chris Kiefer, Thor Magnusson, and Dorien Herremans. 2022. MusIAC: An extensible generative framework for music infilling applications with multi-level control. In *Artificial Intelligence in Music, Sound, Art and Design*. Lecture Notes in Computer Science, Vol. 13221. Springer, 341–356.

[80] Zixun Guo, Jaeyong Kang, and Dorien Herremans. 2023. A domain-knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. AAAI Press, Article 566, 8 pages.

[81] Gaëtan Hadjeres and Léopold Crestel. 2021. The piano inpainting application. *arXiv:2107.05944* (2021).

[82] Gaëtan Hadjeres and Frank Nielsen. 2017. Interactive music generation with positional constraints using anticipation-RNNs. *arXiv:1709.06404* (2017).

[83] Gaëtan Hadjeres, François Pachet, and Frank Nielsen. 2017. DeepBach: A steerable model for Bach chorales generation. In *Proceedings of the International Conference on Machine Learning (ICML'17)*. PMLR, 1362–1371.

[84] Sophia Hager, Kathleen Hablutzel, and Katherine M. Kinnaird. 2024. Generating music with structure using self-similarity as attention. *arXiv:2406.15647* (2024).

[85] Sangjun Han, Hyeongrae Ihm, and Woohyung Lim. 2023. Systematic analysis of music representations from BERT. *arXiv:2306.04628* (2023).

[86] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *Int. Conf. Learn. Represent. (ICLR)*.

[87] Johannes Hentschel, Markus Neuwirth, and Martin Rohrmeier. 2021. The annotated Mozart sonatas: Score, harmony, and cadence. *Transactions of the International Society for Music Information Retrieval* 4, 1 (5 2021), 67–80.

[88] Carlos Hernandez-Olivan and Jose R. Beltran. 2023. Musicaiz: A Python library for symbolic music generation, analysis and visualization. *SoftwareX* 22 (2023), 101365.

[89] Dorien Herremans and Ching-Hua Chuan. 2017. Modeling musical context with Word2vec. In *Proceedings of the International Workshop on Deep Learning and Music*.

[90] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. 2017. A functional taxonomy of music generation systems. *ACM Comput. Surv.* 50, 5 (2017), 30 pages.

[91] Ruben Hillewaere, Bernard Manderick, and Darrell Conklin. 2018. Global feature versus event models for folk song classification. In *International Society for Music Information Retrieval Conference (ISMIR'18)*. 729–734.

[92] Tatsunori Hirai and Shun Sawada. 2019. Melody2vec: Distributed representations of melodic phrases based on melody segmentation. *Journal of Information Processing* 27 (2019), 278–286.

[93] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (11 1997), 1735–1780.

[94] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. 2021. Compound Word Transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conf. Artif. Intell.* 35 (2021), 178–186.

[95] Anna Huang, Monica Dinculescu, Ashish Vaswani, and Douglas Eck. 2018. Visualizing music self-attention. In *Proceedings of the NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language*. 1.

[96] Chu-Ren Huang and Nianwen Xue. 2012. Words without boundaries: Computational approaches to Chinese word segmentation. *Language and Linguistics Compass* 6, 8 (2012), 494–505.
[97] Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. 2017. Counterpoint by convolution. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'17)*.
[98] Cheng-Zhi Anna Huang, David Duvenaud, and Krzysztof Z. 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)*. 241–250.
[99] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music Transformer. In *International Conference on Learning Representations*.
[100] Hen-Hsen Huang, Chuen-Tsai Sun, and Hsin-Hsi Chen. 2010. Classical Chinese sentence segmentation. In *Proceedings of the CIPS-SIGHAN Joint Conference on Chinese Language Processing*.
[101] Yu-Siang Huang and Yi-Hsuan 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)*. 1180–1188.
[102] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. 2021. EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21)*. 318–325.
[103] Ray Jackendoff. 2009. Parallels and nonparallels between language and music. *Music Perception: An Interdisciplinary Journal* 26, 3 (2009), 195–204.
[104] Fatemeh Jamshidi, Gary Pike, Amit Das, and Richard Chapman. 2024. Machine learning techniques in automatic music transcription: A systematic survey. *arXiv:2406.15249* (2024).
[105] Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *J. Artif. Int. Res.* 65, 1 (2019), 675–682.
[106] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, and Juhan Nam. 2019. VirtuosoNet: A hierarchical RNN-based system for modeling expressive piano performance. In *International Society for Music Information Retrieval Conference (ISMIR'19)*. 908–915.
[107] Shulei Ji, Xinyu Yang, and Jing Luo. 2023. A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. *ACM Comput. Surv.* 56, 1 (2023), 39 pages.
[108] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, LAĺlio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, TimothAĺe Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL]
[109] Junyan Jiang, Gus Xia, and Taylor Berg-Kirkpatrick. 2020. Discovering music relations with sequential attention. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA'20)*. Association for Computational Linguistics, Online, 1–5.
[110] Junyan Jiang, Gus G. Xia, Dave B. Carlton, Chris N. Anderson, and Ryan H. 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.
[111] Cong Jin, Yun Tie, Yong Bai, Xin Lv, and Shouxun Liu. 2020. A style-specific music composition neural network. *Neural Process. Lett.* 52, 3 (2020), 1893–1912.
[112] Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics* 48, 1 (2022), 155–205.
[113] Zeqian Ju, Peiling Lu, Xu Tan, Rui Wang, Chen Zhang, Songruoyao Wu, Kejun Zhang, Xiangyang Li, Tao Qin, and Tie-Yan Liu. 2022. TeleMelody: Lyric-to-melody generation with a template-based two-stage method. *arXiv:2109.09617* (2022).
[114] Dan Jurafsky. 2000. *Speech & Language Processing*.
[115] Jaeyong Kang, Soujanya Poria, and Dorien Herremans. 2023. Video2Music: Suitable music generation from videos using an affective multimodal Transformer model. *arXiv:2311.00968* (2023).
[116] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive Transformers with linear attention. In *International Conference on Machine Learning (ICML) (ICML'20)*. JMLR.org, Article 478, 10 pages.
[117] Mathieu Kermarec, Louis Bigo, and Mikaela Keller. 2022. Improving tokenization expressiveness with pitch intervals. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'22): Late-Breaking Demo Session*.
[118] Brett Kessler, Geoffrey Nunberg, and Hinrich Schuetze. 1997. Automatic detection of text genre. *arXiv:cmp-lg/9707002* (1997).

[119] Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, et al. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Minneapolis, Minnesota, 235–249.

[120] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. In *Int. Conf. Learn. Represent. (ICLR)*.

[121] Stephen James Krol, Maria Teresa Llano, and Jon McCormack. 2022. Towards the generation of musical explanations with GPT-3. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*. Springer, 131–147.

[122] Taku 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*. Association for Computational Linguistics, Melbourne, Australia, 66–75.

[123] Adarsh Kumar and Pedro Sarmento. 2023. From words to music: A study of subword tokenization techniques in symbolic music generation. *arXiv:2304.08953* (2023).

[124] Harish Kumar and Balaraman Ravindran. 2019. Polyphonic music composition with LSTM neural networks and reinforcement learning. *arXiv:1902.01973* (2019).

[125] Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech & Language* 6, 3 (1992), 225–242.

[126] Allison Lahnala, Gauri Kambhatla, Jiajun Peng, Matthew Whitehead, Gillian Minnehan, Eric Guldan, Jonathan K. Kummerfeld, Anıl Çamcı, and Rada 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*. 171–186.

[127] Nicolas Lazzari, Andrea Poltronieri, and Valentina Presutti. 2023. Pitchclass2vec: Symbolic music structure segmentation with chord embeddings. *arXiv:2303.15306* (2023).

[128] Hyun Lee, Taehyun Kim, Hyolim Kang, Minjoo Ki, Hyeonchan Hwang, Sharang Han, Seon Joo Kim, et al. 2022. ComMU: Dataset for combinatorial music generation. *Adv. Neural Inf. Process. Syst. (NeurIPS)* 35 (2022), 39103–39114.

[129] Fred Lerdahl. 2012. *Musical Syntax and Its Relation to Linguistic Syntax*. Collège de France.

[130] Fred Lerdahl and Ray S. Jackendoff. 1996. *A Generative Theory of Tonal Music*. MIT Press.

[131] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv:1910.13461* (2019).

[132] Jiajia Li, Lu Yang, Mingni Tang, Cong Chen, Zuchao Li, Ping Wang, and Hai Zhao. 2024. The music maestro or the musically challenged, a massive music evaluation benchmark for large language models. *arXiv:2406.15885* (2024).

[133] Shuyu Li and Yunsick Sung. 2023. MelodyDiffusion: Chord-conditioned melody generation using a Transformer-based diffusion model. *Mathematics* 11, 8 (2023).

[134] Shuyu Li and Yunsick Sung. 2023. MRBERT: Pre-training of melody and rhythm for automatic music generation. *Mathematics* 11, 4 (2023), 798.

[135] Yuqiang Li, Shengchen Li, and George Fazekas. 2023. An comparative analysis of different pitch and metrical grid encoding methods in the task of sequential music generation. *arXiv:2301.13383* (2023).

[136] Yuqiang Li, Shengchen Li, and George Fazekas. 2023. Pitch class and octave-based pitch embedding training strategies for symbolic music generation. In *International Symposium on Computer Music Multidisciplinary Research (CMMR)*. Zenodo, Tokyo, Japan, 86–97.

[137] Yang Li and Tao Yang. 2018. Word embedding for understanding natural language: A survey. In *Guide to Big Data Applications*. Springer International Publishing, Cham, 83–104.

[138] Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. 2020. PiRhDy: Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music. In *Proceedings of the 28th ACM International Conference on Multimedia (Seattle, WA, USA) (MM'20)*. Association for Computing Machinery, New York, NY, USA, 574–582.

[139] Xiao Liang, Zijian Zhao, Weichao Zeng, Yutong He, Fupeng He, Yiyi Wang, and Chengying Gao. 2024. PianoBART: Symbolic piano music generation and understanding with large-scale pre-training. *arXiv:2407.03361* (2024).

[140] Yaobo Liang, Nan Duan, Yeyun Gong, NingWu, Fenfei Guo,Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 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, 6008–6018.

[141] David 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 (1997), 1–28.

[142] Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M. Saiful 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, 4190–4200.

[143] Chien-Hung Liu and Chuan-Kang Ting. 2017. Computational intelligence in music composition: A survey. *IEEE Transactions on Emerging Topics in Computational Intelligence* 1, 1 (2017), 2–15.

[144] Jiafeng Liu, Yuanliang Dong, Zehua Cheng, Xinran Zhang, Xiaobing Li, Feng Yu, and Maosong Sun. 2022. Symphony generation with permutation invariant language model. In *International Society for Music Information Retrieval Conference (ISMIR'22)*.

[145] Qi Liu, Matt J. Kusner, and Phil Blunsom. 2020. A survey on contextual embeddings. *arXiv:2003.07278* (2020).

[146] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692* (2019).

[147] Antoine Liutkus, Ondřej Cífka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gael Richard. 2021. Relative positional encoding for Transformers with linear complexity. In *International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 7067–7079.

[148] Gabriel Loiseau, Mikaela Keller, and Louis Bigo. 2021. What musical knowledge does self-attention learn? In *Proceedings of the 2nd Workshop on NLP for Music and Spoken Audio (NLP4MusA'21)*. Association for Computational Linguistics, Online, 6–10.

[149] Elias Lousseief and Bob Sturm. 2019. MahlerNet: Unbounded orchestral music with neural networks. In *Proceedings of the 2019 Nordic Sound and Music Computing Conference and the 2019 Interactive Sonification Workshop*. 57–63.

[150] Peiling Lu, Xin Xu, Chenfei Kang, Botao Yu, Chengyi Xing, Xu Tan, and Jiang Bian. 2023. MuseCoco: Generating symbolic music from text. arXiv:2306.00110 [cs.SD]

[151] Jing Luo, Xinyu Yang, and Dorien Herremans. 2024. BandControlNet: Parallel Transformers-based steerable popular music generation with fine-grained spatiotemporal features. *arXiv:2407.10462* (2024).

[152] Sephora Madjiheurem, Lizhen Qu, and Christian Walder. 2016. Chord2vec: Learning musical chord embeddings. In *Proceedings of the Constructive Machine Learning Workshop at NIPS*.

[153] Dimos Makris, Kat R. Agres, and Dorien Herremans. 2021. Generating lead sheets with affect: A novel conditional seq2seq framework. In *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN'21)*.

[154] Dimos Makris, Guo Zixun, Maximos Kaliakatsos-Papakostas, and Dorien Herremans. 2022. Conditional drums generation using compound word representations. In *Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art, and Design*. Springer, 179–194.

[155] Martin E. Malandro. 2024. Composer's Assistant 2: Interactive multi-track MIDI infilling with fine-grained user control. arXiv:2407.14700 [cs.SD]

[156] Kristen Masada and Razvan C. Bunescu. 2017. Chord recognition in symbolic music using semi-Markov conditional random fields. In *International Society for Music Information Retrieval Conference (ISMIR)*. 272–278.

[157] Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Natural Language Learning*. 188–191.

[158] Cory McKay, Julie Cumming, and Ichiro 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.

[159] Jan Melechovsky, Abhinaba Roy, and Dorien Herremans. 2024. MidiCaps: A large-scale MIDI dataset with text captions. *arXiv:2406.02255* (2024).

[160] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 32. Curran Associates, Inc.

[161] Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. *arXiv:2112.10508* (2021).

[162] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).

[163] Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao. 2023. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. In *International Society for Music Information Retrieval Conference (ISMIR'23)*.

[164] Aashiq Muhamed, Liang Li, Xingjian Shi, Suri Yaddanapudi, Wayne Chi, Dylan Jackson, Rahul Suresh, Zachary C. Lipton, and Alex J. Smola. 2021. Symbolic music generation with Transformer-GANs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 35, 1 (2021), 408–417.

[165] Néstor Nápoles López, Claire Arthur, and Ichiro Fujinaga. 2019. Key-finding based on a hidden Markov model and key profiles. In *Proceedings of the 6th International Conference on Digital Libraries for Musicology (DLfM'19)*. 33–37.
[166] Eugene Narmour. 1990. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press.
[167] Pedro Neves, Jose Fornari, and Joao Florindo. 2022. Generating music with sentiment using Transformer-GANs. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'22)*.
[168] Seol-Hyun Noh. 2021. Analysis of gradient vanishing of RNNs and performance comparison. *Information* 12, 11 (2021).
[169] Mitsunori Ogihara and Tao Li. 2008. N-gram chord profiles for composer style representation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'08)*. 671–676.
[170] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. 2018. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications* 32, 4 (2018), 955–967.
[171] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael 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*. Association for Computational Linguistics, Minneapolis, Minnesota, 48–53.
[172] David D. Palmer. 2000. Tokenisation and sentence segmentation. In *Handbook of Natural Language Processing*. CRC Press, 11.
[173] Saebyul Park, Eunjin Choi, Jeounghoon Kim, and Juhan Nam. 2024. Mel2Word: A text-based melody representation for symbolic music analysis. *Music & Science* 7 (2024).
[174] Christine Payne. 2019. MuseNet.
[175] Marcus T. Pearce. 2018. Statistical learning and probabilistic prediction in music cognition: Mechanisms of stylistic enculturation. *Annals of the New York Academy of Sciences* 1423, 1 (2018), 378–395.
[176] Yifan Peng, Shankai Yan, and Zhiyong 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*. 58–65.
[177] Juan I. Perotti and Orlando V. Billoni. 2020. On the emergence of Zipf's law in music. *Physica A: Statistical Mechanics and Its Applications* 549 (2020), 124309.
[178] E. Pollastri and G. Simoncelli. 2001. Classification of melodies by composer with hidden Markov models. In *International Conference on WEB Delivering of Music*. 88–95.
[179] Friedemann Pulvermüller and Ramin Assadollahi. 2007. Grammar or serial order?: Discrete combinatorial brain mechanisms reflected by the syntactic mismatch negativity. *Journal of Cognitive Neuroscience* 19, 6 (2007), 971–980.
[180] Yang Qin, Huiming Xie, Shuxue Ding, Benying Tan, Yujie Li, Bin Zhao, and Mao Ye. 2022. Bar Transformer: A hierarchical model for learning long-term structure and generating impressive pop music. *Applied Intelligence* 53, 9 (2022), 10130–10148.
[181] Lvyang Qiu, Shuyu Li, and Yunsick Sung. 2021. DBTMPE: Deep bidirectional Transformers-based masked predictive encoder approach for music genre classification. *Mathematics* 9, 5 (2021).
[182] Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xinrun Du, Shuyue Guo, Yiming Liang, Yizhi Li, Shangda Wu, Junting Zhou, Tianyu Zheng, Ziyang Ma, Fengze Han, Wei Xue, Gus Xia, Emmanouil Benetos, Xiang Yue, Chenghua Lin, Xu Tan, Stephen W. Huang, Wenhu Chen, Jie Fu, and Ge Zhang. 2024. MuPT: A Generative Symbolic Music Pretrained Transformer. arXiv:2404.06393 [cs.SD]
[183] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
[184] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
[185] Colin Raffel. 2016. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*.
[186] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *J. Mach. Learn. Res.* 21, 1 (2020), 67 pages.
[187] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning (ICML'21)*. PMLR, 8821–8831.
[188] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *arXiv:1908.10084* (2019).

[189] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. PopMAG: Pop music accompaniment generation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1198–1206.
[190] C. Roads and Paul Wieneke. 1979. Grammars as representations for music. *Computer Music Journal* 3, 1 (1979), 48–55.
[191] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research)*. PMLR, 4364–4373.
[192] Martin Rohrmeier. 2011. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music* 5, 1 (2011), 35–53.
[193] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 6088 (1986), 533–536.
[194] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv:1910.01108* (2020).
[195] Pedro Sarmento, Adarsh Kumar, C. J. Carr, Zack Zukowski, Mathieu Barthet, and Yi-Hsuan Yang. 2021. DadaGP: A dataset of tokenized GuitarPro songs for sequence models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21)*.
[196] Pedro Sarmento, Adarsh Kumar, Yu-Hua Chen, C. J. Carr, Zack Zukowski, and Mathieu Barthet. 2023. GTR-CTRL: Instrument and genre conditioning for guitar-focused music generation with Transformers. In *Artificial Intelligence in Music, Sound, Art and Design*. Lecture Notes in Computer Science, Vol. 13988. Springer, 260–275.
[197] Pedro Sarmento, Adarsh Kumar, Dekun Xie, C. J. Carr, Zack Zukowski, and Mathieu Barthet. 2023. ShredGP: Guitarist style-conditioned tablature generation with Transformers. In *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research*. 112–121.
[198] Helmut Schaffrath. 1995. *The Essen Folksong Collection*. Center for Computer Assisted Research in the Humanities.
[199] Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'12)*. 5149–5152.
[200] David R. W. Sears, Andreas Arzt, Harald Frostel, Reinhard Sonnleitner, and Gerhard Widmer. 2017. Modeling harmony with skip-grams. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'17)*.
[201] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany, 1715–1725.
[202] Marc Serra-Peralta, Joan Serrà, and Álvaro Corral. 2021. Heaps' law and vocabulary richness in the history of classical music harmony. *EPJ Data Science* 10, 1 (2021), 40.
[203] Senturk Sertan and Parag Chordia. 2011. Modeling melodic improvisation in Turkish folk music using variable-length Markov models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'11)*. 269–274.
[204] Zhexu Shen, Liang Yang, Zhihan Yang, and Hongfei 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*. 540–544.
[205] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Müller, and Yi-Hsuan Yang. 2023. Theme Transformer: Symbolic music generation with theme-conditioned Transformer. *IEEE Transactions on Multimedia* 25 (2023), 3495–3508.
[206] Yangyang Shu, Haiming Xu, Ziqin Zhou, Anton van den Hengel, and Lingqiao Liu. 2024. MuseBarControl: Enhancing fine-grained control in symbolic music generation through pre-training and counterfactual loss. *arXiv:2407.04331* (2024).
[207] Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology* 60, 3 (2009), 538–556.
[208] Mark J. Steedman. 1984. A generative grammar for jazz chord sequences. *Music Perception* 2, 1 (10 1984), 52–77.
[209] Bob L. Sturm, João Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. 2016. Music transcription modelling and composition using deep learning. *arXiv:1604.08723* (2016).
[210] Serkan Sulun, Matthew E. P. Davies, and Paula Viana. 2022. Symbolic music generation conditioned on continuous-valued emotions. *IEEE Access* 10 (2022), 44617–44626.
[211] Jingjing Tang, Geraint Wiggins, and Gyorgy Fazekas. 2023. Reconstructing human expressiveness in piano performances with a Transformer network. *arXiv:2306.06040* (2023).
[212] Hanqing Tao, Shiwei Tong, Hongke Zhao, Tong Xu, Binbin Jin, and Qi Liu. 2019. A radical-aware attention-based model for Chinese text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 33, 01 (2019), 5125–5132.
[213] Renaud Bougueng Tchemeube, Jeffrey Ens, Cale Plut, Philippe Pasquier, Maryam Safi, Yvan Grabit, and Jean-Baptiste Rolland. 2023. Evaluating human-AI interaction via usability, user experience and acceptance measures for

MMM-C: A creative AI system for music composition. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 10 pages.

[214] Jinhao Tian, Zuchao Li, Jiajia Li, and Ping Wang. 2024. N-gram unsupervised compoundation and feature injection for better symbolic music understanding. In *Proceedings of the AAAI Conf. Artif. Intell.* 38, 14 (Mar. 2024), 15364–15372.

[215] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288* (2023).

[216] Nicholas Trieu and R. Keller. 2018. JazzGAN: Improvising with generative adversarial networks. In *MUME Workshop*.

[217] Meliksah Turker, Alara Dirik, and Pinar Yanardag. 2022. MIDISpace: Finding linear directions in latent space for music generation. In *Proceedings of the 14th Conference on Creativity and Cognition (C&C'22)*. Association for Computing Machinery, New York, NY, USA, 420–427.

[218] Andries Van Der Merwe and Walter Schulze. 2011. Music generation with Markov models. *IEEE MultiMedia* 18, 3 (2011), 78–85.

[219] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 30. Curran Associates, Inc., 6000–6010.

[220] Barry Lloyd Vercoe. 2001. Folk music classification using hidden Markov models. In *Proceedings of the International Conference on Artificial Intelligence*, Vol. 6.

[221] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan 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, 5797–5808.

[222] Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. 2023. FIGARO: Controllable music generation using learned and expert features. In *Int. Conf. Learn. Represent. (ICLR)*.

[223] Nils L. Wallin, Bjorn Merker, and Steven Brown. 2001. *The Origins of Music*. MIT press.

[224] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*. Association for Computational Linguistics, Brussels, Belgium, 353–355.

[225] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C.-C. Jay Kuo. 2019. Evaluating word embedding models: Methods and experimental results. *APSIPA Transactions on Signal and Information Processing* 8 (2019), 19.

[226] Lei Wang, Ziyi Zhao, Hanwei Liu, Junwei Pang, Yi Qin, and Qidi Wu. 2023. A review of intelligent music generation systems. *arXiv:2211.09124* (2023).

[227] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, and Gus Xia. 2020. POP909: A pop-song dataset for music arrangement generation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'20)*. ISMIR, Montreal, Canada, 38–45.

[228] Ziyu Wang and Gus Xia. 2021. MuseBERT: Pre-training of music representation for music understanding and controllable generation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'21)*. 722–729.

[229] Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review* 55, 7 (2022), 5731–5780.

[230] Joseph Weizenbaum. 1966. ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9, 1 (1 1966), 36–45.

[231] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. CCNet: Extracting high quality monolingual datasets from web crawl data. *arXiv:1911.00359* (2019).

[232] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, 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, 38–45.

[233] Jacek Wołkowicz and Vlado 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*. Lecture Notes in Computer Science, Vol. 7310. Springer, 230–241.

[234] Jacek Wołkowicz, Zbigniew Kulka, and Vlado Kešelj. 2008. N-gram-based approach to composer recognition. *Archives of Acoustics* 33, 1 (2008).

[235] Kam-Fai Wong, Wenjie Li, Ruifeng Xu, and Zheng-Sheng Zhang. 2022. *Introduction to Chinese Natural Language Processing*. Springer.

[236] Shangda Wu and Maosong Sun. 2023. Exploring the efficacy of pre-trained checkpoints in text-to-music generation task. In *Proceedings of the AAAI-23 Workshop on Creative AI across Modalities*.

[237] Shangda Wu, Yashan Wang, Xiaobing Li, Feng Yu, and Maosong Sun. 2024. MelodyT5: A unified score-to-score Transformer for symbolic music processing. *arXiv:2407.02277* (2024).

[238] Shangda Wu, Dingyao Yu, Xu Tan, and Maosong Sun. 2023. CLaMP: Contrastive language-music pre-training for cross-modal symbolic music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'23)*.

[239] Shih-Lun Wu and Yi-Hsuan Yang. 2020. The Jazz Transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'20)*. 142–149.

[240] Shih-Lun Wu and Yi-Hsuan Yang. 2023. Compose & Embellish: Well-structured piano performance generation via a two-stage approach. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'23)*. 1–5.

[241] Shih-Lun Wu and Yi-Hsuan 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 (2023), 1953–1967.

[242] Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 62–69.

[243] Weihan Xu, Julian McAuley, Shlomo Dubnov, and Hao-Wen Dong. 2023. Equipping pretrained unconditional music Transformers with instrument and genre controls. In *2023 IEEE International Conference on Big Data (BigData)*. 4512–4517.

[244] Li-Chia Yang and Alexander Lerch. 2020. On the evaluation of generative models in music. *Neural Computing and Applications* 32, 9 (2020), 4773–4784.

[245] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 32. Curran Associates, Inc., Vancouver, Canada.

[246] Georgios N. Yannakakis and Héctor P. Martínez. 2015. Ratings are overrated! *Frontiers in ICT* 2 (2015).

[247] Botao Yu, Peiling Lu, Rui Wang, Wei Hu, Xu Tan, Wei Ye, Shikun Zhang, Tao Qin, and Tie-Yan Liu. 2022. Museformer: Transformer with fine-and coarse-grained attention for music generation. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vol. 35. 1376–1388.

[248] Yi Yu, Abhishek Srivastava, and Simon Canales. 2021. Conditional LSTM-GAN for melody generation from lyrics. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 1, Article 35 (4 2021), 20 pages.

[249] Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, et al. 2024. ChatMusician: Understanding and generating music intrinsically with LLM. *arXiv:2402.16153* (2024).

[250] Lawrence M. Zbikowski. 2009. Music, language, and multimodal metaphor. *Multimodal Metaphor* (2009), 359–381.

[251] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan 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. 791–800.

[252] Huan Zhang, Emmanouil Karystinaios, Simon Dixon, Gerhard Widmer, and Carlos Eduardo Cancino-Chacón. 2023. Symbolic music representations for classification tasks: A systematic evaluation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'23)*.

[253] Li Zhang and Chris Callison-Burch. 2023. Language models are drummers: Drum composition with natural language pre-training. In *Proceedings of the AAAI-23 Workshop on Creative AI across Modalities*.

[254] Liumei Zhang and Fanzhi Jiang. 2021. Visualizing symbolic music via textualization: An empirical study on Chinese traditional folk music. In *Mobile Multimedia Communications*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 394. Springer, 647–662.

[255] Ning Zhang. 2020. Learning adversarial Transformer for symbolic music generation. *IEEE Transactions on Neural Networks and Learning Systems* 34, 4 (2020), 1754–1763.

[256] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.* 15, 2 (1 2024), 1–38.

[257] Jing Zhao, David Taniar, Kiki Adhinugraha, Vishnu Monn Baskaran, and KokSheik Wong. 2023. Multi-mmlg: A novel framework of extracting multiple main melodies from MIDI files. *Neural Computing and Applications* 35, 30 (2023), 22687–22704.

[258] Jing Zhao, KokSheik Wong, Vishnu Monn Baskaran, Kiki Adhinugraha, and David Taniar. 2023. Computational music: Analysis of music forms. In *Computational Science and Its Applications (ICCSA)* (Athens, Greece). Springer-Verlag, Berlin, Heidelberg, 366–384.

[259] Jingwei Zhao, Gus Xia, and Ye Wang. 2023. AccoMontage-3: Full-band accompaniment arrangement via sequential style transfer and multi-track function prior. *arXiv:2310.16334* (2023).
[260] Jingwei Zhao, Gus Xia, and Ye Wang. 2023. Q&A: Query-based representation learning for multi-track symbolic music re-arrangement. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*.
[261] Jiuyang Zhou, Hong Zhu, and Xingping Wang. 2023. Choir Transformer: Generating polyphonic music with relative attention on Transformer. *arXiv:2308.02531* (2023).
[262] Ziya Zhou, Yuhang Wu, Zhiyue Wu, Xinyue Zhang, Ruibin Yuan, Yinghao Ma, Lu Wang, Emmanouil Benetos, Wei Xue, and Yike Guo. 2024. Can LLMs "Reason" in music? An evaluation of LLMs' capability of music understanding and generation. *arXiv:2407.21531* (2024).
[263] Hongyuan Zhu, Qi Liu, Nicholas Jing Yuan, Chuan Qin, Jiawei Li, Kun Zhang, Guang Zhou, Furu Wei, Yuanchun Xu, and Enhong Chen. 2018. XiaoIce band: A melody and arrangement generation framework for pop music. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*. 2837–2846.
[264] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *arXiv:2308.07633* (2023).
[265] Yueyue Zhu, Jared Baca, Banafsheh Rekabdar, and Reza Rawassizadeh. 2023. A survey of AI music generation tools and models. *arXiv:2308.12982* (2023).
[266] Guo Zixun, Dimos Makris, and Dorien Herremans. 2021. Hierarchical recurrent neural networks for conditional melody generation with long-term structure. In *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN'21)*.

Received 26 February 2024; revised 15 October 2024; accepted 6 January 2025