The Harmonic Memory: a Knowledge Graph of harmonic patterns as a trustworthy framework for computational creativity

Jacopo de Berardinis
Albert Meroño-Peñuela
jacopo.deberardinis@kcl.ac.uk
King's College London
London, UK

Andrea Poltronieri
Valentina Presutti
andrea.poltronieri2@unibo.it
University of Bologna
Bologna, Italy

ABSTRACT

Computationally creative systems for music have recently achieved impressive results, fuelled by progress in generative machine learning. However, black-box approaches have raised fundamental concerns for ethics, accountability, explainability, and musical plausibility. To enable trustworthy machine creativity, we introduce the *Harmonic Memory*, a Knowledge Graph (KG) of harmonic patterns extracted from a large and heterogeneous musical corpus. By leveraging a cognitive model of tonal harmony, chord progressions are segmented into meaningful structures, and patterns emerge from their comparison via harmonic similarity. Akin to a music memory, the KG holds temporal connections between consecutive patterns, as well as salient similarity relationships. After demonstrating the validity of our choices, we provide examples of how this design enables novel pathways for combinational creativity. The memory provides a fully accountable and explainable framework to inspire and support creative professionals – allowing for the discovery of progressions consistent with given criteria, the recomposition of harmonic sections, but also the co-creation of new progressions.

CCS CONCEPTS

• **Applied computing → Sound and music computing**; • **Computing methodologies → Knowledge representation and reasoning**; *Ontology engineering*.

KEYWORDS

computational creativity, knowledge graphs, music technology

ACM Reference Format:
Jacopo de Berardinis, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. 2023. The Harmonic Memory: a Knowledge Graph of harmonic patterns as a trustworthy framework for computational creativity. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3543507.3587428

1 INTRODUCTION

Creativity has been defined as the ability to come up with new, surprising, and valuable ideas or artifacts [4]. These can be abstract concepts, scientific theories, solutions to real-world problems, but also new designs and artworks. In her seminal work, Boden categorised creativity into three types: (i) *exploratory*, where new ideas are generated by exploration of a space of concepts; (ii) *combinational*, which enables the creation of new ideas through the combination of familiar ones; and (iii) *transformational*, where the "the rules" governing a space are challenged and transformed, to generate new kinds of ideas. A computational creativity theory was also formulated by Colton et al., to describe creative and generative acts (FACE model) and their potential for impact (IDEA model).

Attempts at formalising human creativity date back to the ancient Greeks, and remained up to and beyond Mozart with the "*Dice Game*" and Ada Lovelace – speculating that the "calculating engine" might compose elaborate and scientific pieces of music of any degree of complexity. Since then, creativity, creative reasoning and creative problem solving have been extensively researched in cognitive [5] and computational sciences [24]. A simple definition of a computationally creative system is that of a model capable to perform "generative acts" that *create* artefacts, concepts, or provide an aesthetic *evaluation* for the generated outputs [23, 40]. By harnessing recent advancements in machine learning, a variety of systems have already been implemented across several domains. Examples include computational systems for material discovery [9], molecular design [66], and more broadly, for virtual laboratories [36]; but also models for generating textual artefacts [52], images [58, 61], and even recipes [62] from a variety of prompts.

In the music domain, data-driven generative systems based on deep learning methods have achieved impressive results on symbolic music [7], and they can also produce realistic outputs when trained on the raw audio [21]. The variety of computationally creative methods for music is quite broad and diversified, and has already enabled the exploration of novel forms of artistic co-creation [33]. These range from the automatic generation, completion, and alteration of chord progressions and melodies, to the creation of mashups, and audio snippets from textual prompts [2]. Due to their success, some of these systems have already been integrated into commercial software, such as *Aiva*[1] and *Amper* [2] – allowing users to generate full music pieces based on their desiderata.

1.1 Fundamental concerns of music AI systems

Nonetheless, having a system that can fully generate realistic music raises ethical concerns – especially when those systems are made commercial and can potentially replace artists, rather than augmenting their possibilities [65]. Indeed, research can open highly

[1]https://www.aiva.ai
[2]https://www.ampermusic.com

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3587428

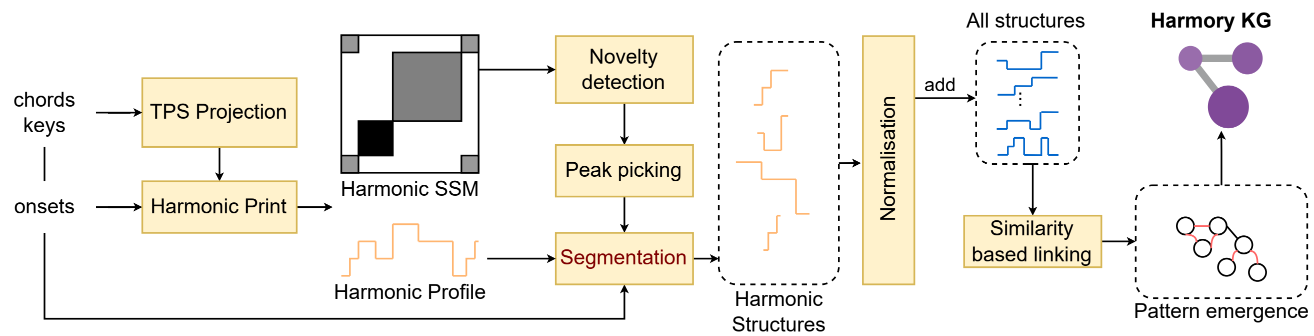

Figure 1: Overview of the main steps for the creation of Harmory, from the encoding of chord progressions in the Tonal Pitch Space (TPS) and their segmentation, to the emergence of harmonic patterns through similarity and the creation of the KG.

lucrative business opportunities given the low cost of non-human musicians and "*their inability to organise in unions to protest against unfair treatment*" [48].

In addition, computationally creative models that fully learn music representations from the data by maximising a learning objective (e.g. autoregressive, masked prediction, generative modelling) are often criticised for lacking *accountability*, *explainability*, and *musical plausibility*. The former is related to the challenge of keeping track of where the model picks up while generating new musical content. As the model is unaware of its influences while composing, this may prevent giving recognition to real artists, which has direct implication on copyright and revenue sharing [22]. Similarly, the lack of explainability represents a technological barrier for users, as there is little or no understanding of the creative process underneath. Explainability is a desirable component for computationally creative systems, as it facilitates the interaction with artists, and particularly, the ability to control/steer the system based on domain knowledge [6, 8]. Finally, the "creative space" learned by data-driven systems is often criticised by musicologists and music experts in regard to musical plausibility [30], meaning that, solutions generated from these models may violate common notions of music theory. This fundamentally hampers a potential dialogue and synergies between music experts and AI researchers.

In sum, most music AI systems cannot yet be deemed trustworthy by design (accountability, explainability, ethics, etc.) [26], which raises serious concerns related to their large scale adoption.

1.2 Our contribution

Instead of replacing artists, we believe that research should focus on leveraging the generative capabilities of these systems to design novel solutions that can support, enhance and augment the creative potential of composers as a human-machine collaboration [11].

Inspired by research in music psychology [37], here we present the **Harmonic Memory** (Harmory) – a Knowledge Graph (KG) of harmonic patterns aimed to support creative applications in a fully transparent, accountable, and musically plausible way.

We leverage a cognitive model of Western tonal harmony to project chord progressions into a musically meaningful space, and signal processing methods to segment the resulting sequences into meaningful harmonic structures. The latter are then compared with each other, across all progressions and via harmonic similarity, to reveal common/recurring harmonic patterns. Finally, a KG is created to semantically establish relationships between patterns, based on: (i) *temporal links*, connecting two patterns if they are observed consecutively in the same progression; and (ii) *similarity links* among highly-similar patterns. Trivially, identical patterns do not need connections, as they are mapped to the same entity.

By traversing the KG, and moving across patterns via temporal and similarity links, new progressions can be created in a combinational settings; but also, unexpected and surprising relationships can be found among pieces and composers of different genre, style, and historical period. This is also enabled by the scale and diversity of Harmory, which is built from ChoCo [16] – the largest existing collection of harmonic annotations.

Our main contributions can be summarised as follows.

- The Harmonic Memory (Harmory), a large, diversified, and musically meaningful KG of harmonic patterns aimed to support applications of trustworthy machine creativity.
- Underpinning Harmory, we also contribute and validate empirically: (i) a novel method for harmonic structure analysis in the symbolic domain, that leverages cognitive and musicological models of tonal harmony; (ii) and a state of the art algorithm for harmonic similarity.
- Examples of possible applications for trustworthy machine creativity implemented on top of Harmory, focusing on knowledge discovery and human-machine chord generation.

2 RELATED WORK

To the best of our knowledge, most machine learning systems are *explorative*. Starting from different prompts, such as a priming music to continue, an incomplete passage, or a textual query, these models can generate convincing outputs by sampling from the learned distribution. These include methods based on recurrent [64], self-attention [34], and convolutional neural networks [32]. Instead, current *combinational* systems are dominated by variational autoencoders, which can create new ideas by interpolating between two musical passages in a latent space [60]. *Transformative* approaches for music have been implemented by "hacking" the former methods based on the idea of brain transplant, to provide additional artistic stimulation [12]. These range from gentler interventions mixing up corpora, to splicing neural networks, jointly training with interference, and Frankensteinian hybrid models [67].

As pointed out before, most of these works lack trustworthy features to support and protect creative professionals. Recently, *Explainable Computational Creativity* (XCC) systems have been proposed, to promote a bidirectional interaction between system and user [44]. This interaction is communicative, enabling the exchange of decisions and ideas in a format that can be understood by both humans and machines. Examples of explainable systems also include [14] – presenting a real-time human-machine interaction for artwork creation: the system provides explanations for its decisions, while users can guide the creative process.

Semantic Web technologies have also been used to make creative systems more explainable. An example is [55], which proposes a system for creating innovative food combinations using a knowledge graph that describes compounds and ingredients. However, to the best of our knowledge, no such systems have been proposed in the musical domain. A notable exception is the work by Meerwaldt et al., enabling the generation of mashups by leveraging Semantic Web technologies for machine creativity [40]. Our work differs substantially in the broader intent and creative applications it enables, the musicological and cognitive basis, the scope/granularity of the interconnected musical content (patterns vs full pieces).

3 HARMORY: THE HARMONIC MEMORY

The main steps for the creation of Harmory are illustrated in Figure 1, and encompass four stages: (i) projection of harmonic sequences in the Tonal Pitch Space; (ii) novelty-based segmentation of harmonic sequences; (iii) pattern identification through similarity-based linking of harmonic segments; and (iii) KG creation.

Our workflow is defined from the harmonic analysis of a piece, which contains a sequence of *chords* in Harte notation [31], their *onsets*, and the associated local *keys*. Formally, let $\mathbf{c} = \{c_1, \ldots, c_N\}$ denote a chord sequence of length N, where each chord figure c_i is drawn from the Harte chordal set $\mathcal{H}$. Similarly, $\mathbf{k} = \{k_1, \ldots, k_N\}$ denotes the corresponding local keys of $\mathbf{c}$, s.t. each k_i is a tonic-mode tuple defined from $\mathbb{T} \times \mathbb{M}$, where $\mathbb{T} = \{Ab, A, A\#, Bb, \ldots, G\#\}$ is the set of all possible tonic notes, and $\mathbb{M} = \{\text{major}, \text{dorian}, \ldots, \text{locrian}\}$ is the set of all possible modes in Western tonal music.

For simplicity, chords are expected to be temporally aligned with their onsets, meaning that c_i ends when c_{i+1} starts, $\forall i \in N-1$. Hence, *onsets* are defined as a $(N+1)$-th dimensional vector $\mathbf{t} \in \mathbb{R}^{N+1}$ to compensate for the end time of the last chord (t_{N+1} is the end of c_N). Onsets are given in seconds for harmonic analyses on audio music; or as global beats for symbolic music. For example, $\mathbf{c} =$ [`G`, `B:min`, `E:min7`, ...], $\mathbf{k} =$ [(`G`, major), (`G`, major), (`G`, major), ...], and $\mathbf{t} = [1, 3, 5, \ldots]$ are the first three occurrences of such vectors for a "*A Day in the Life*" by The Beatles.

3.1 Encoding chords in the Tonal Pitch Space

Given a harmonic analysis $\mathbf{H} = \{\mathbf{c}, \mathbf{k}, \mathbf{t}\}$, the first step is to encode $\mathbf{c}$ and $\mathbf{k}$ as a numerical stream, so as to allow the processing of similarity/distance operations. This is necessary because chords ($\mathbf{c}$) and tonalities ($\mathbf{k}$) are complex elements to process, and come in symbolic format. More specifically, a chord label is a convention for describing intervals built on a root note. For example, the label of a C major seventh chord (`C:maj7`) represents the intervals of a major quadriad with a minor seventh built on the note C, which is equivalent to the note set $\{C, E, G, Bb\}$. Also, the harmonic function of a chord is contextual to the global (and local) key [1].

One option here is to leverage Representation Learning methods on symbolic music to learn harmonic embeddings from a large corpus of chord sequences [39, 41]. These include static embedding methods, such as Word2Vec [47] and Glove [53], as well as sequence models for contextualised representations, such as ELMo [54] and BERT [20] – which have proved their efficacy on a variety of natural language processing tasks. Nonetheless, in the music domain, representation learning methods have recently started to gain success for audio music [35], whereas little attention has been given to symbolic music. This is exacerbated by the challenge of finding musicological interpretability of the resulting embeddings, requiring new probing and evaluation methods for music [30].

We aim for an encoding of harmony that is well established, perceptually and musicologically plausible, and explainable by design. Hence, we rely on the Tonal Pitch Space [42] – a cognitive model of tonality used in music psychology and computational musicology.

The tonal pitch space

The *Tonal Pitch Space* (TPS) model [42] provides a scoring mechanism that predicts the proximity between musical chords. It is based on the Generative Theory of Tonal Music [43] and designed to make explicit music theoretical and cognitive intuitions about tonal organisation. The model works by comparing any possible chord to an arbitrary key, by means of the *basic space*. The basic space is constituted by five different levels, ordered from the most stable to the least stable: (i) the *Root level*; (ii) the *Fifths level*; (iii) the *Triadic level*; (iv) the *Diatonic level*; and (v) the *Chromatic level*.

Each level holds one or more notes, indexed from 0 (C) to 11 (B). The *Root* level holds the root of a chord (0 for C-major), while the *Fifths* level adds the fifth (0, 7 for C-major). The *Triadic* level has all the notes in the chord (0, 4, 7 for C major). The *Diatonic* level depends on the chord's key as it holds all the notes of the diatonic scale of the key (0, 2, 4, 5, 7, 9, 11 for the C major key). Finally, the *Chromatic* level holds the chromatic scale (0-11).

The distance between two chords c_i, c_j in keys k_i, k_j is calculated using the *basic spaces* of the chords. The basic space is set to match the key of the pieces (level *iv*), and their levels (*i-iii*) are adapted to match the chords to be compared. The Chord distance rule is applied to calculate the distance. The Chord distance rule is defined as $d(x, y) = j + k$, where $d(x, y)$ is the distance between chord x and chord y; j is the minimum number of Circle-of-Fifths rule applications to shift x into y, and k is the number of non-common pitch classes divided by 2 in the levels (*i-iv*) of the basic spaces of x and y. The Circle-of-Fifths rule consists in moving the levels (*i-iii*) four steps to the right or left on level *iv*.

For each comparison between two chords, the TPS returns a value in $[0, 13]$. TPS has been demonstrated to be sound both musicologically and perceptually [18, 19], and in this work, it is used to encode and compare chord-key pairs.

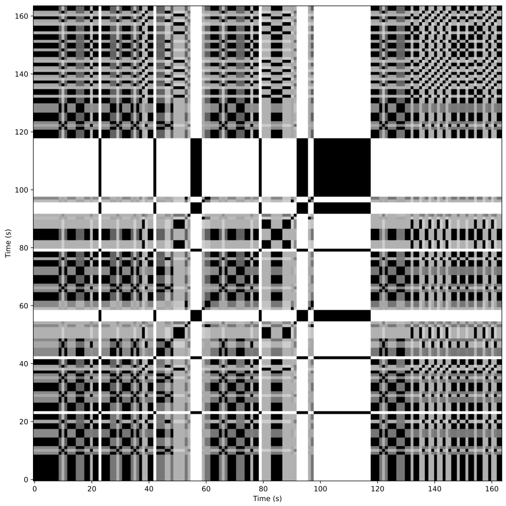

Figure 2: Example of Harmonic SSM resulting from the application of Equation 1 on the TPS signal of "*Crazy Little Things Called Love*" by Queen, using a sampling rate $f_s = 1$. Four main block-like structures are visible, correlating with the musical form of the piece. Smaller, nested harmonic structures of lower granularity are observed withing these blocks.

3.2 Novelty-based harmonic segmentation

The projection of chord-key pairs (c_i, k_i) in the TPS is a fundamental requirement to perform harmonic segmentation. First, the given harmonic annotation $\mathbf{H}$ is used to sample a signal $\mathbf{X}$ of length $d = t_{N+1} \cdot f_s$, where harmonic observation (c_i, k_i) is consecutively repeated $t_i \cdot f_s$ times (its duration), according to a sampling rate f_s. Each element $x_i \in \mathbf{X}$ now encodes an input for the TPS model, containing the harmonic content at the i-th sample.

The resulting signal allows for the computation of two harmonic descriptors, i.e., the *Harmonic Profile* (or TPS time series), and the *Harmonic self-similarity matrix* (SSM) – the entry point for segmentation. The former is defined as a vector $\mathbf{q} \in \mathbb{R}^d$ $s.t.$ $q_i = \texttt{tps}(x_i, k_1)$, holding the TPS distance between each harmonic observation x_i and the global key k_1 of the piece (assumed as the first key occurrence). Similarly, the Harmonic SSM is a matrix $\mathbf{S} \in \mathbb{R}^{d \times d}$ s.t.

$$\mathbf{S}(n, m) = 1 - \frac{\texttt{tps}(x_n, x_m)}{13}, \tag{1}$$

where $x_i \in \mathbf{X}$ is a column vector; $n, m \in [0 : d - 1]$; 13 is a normalisation factor (the maximum TPS value); and the subtraction from 1 is used to obtain a similarity score from a distance measure.

Self-similarity matrices have been extensively used for structure analysis, due to their ability to reveal nested structural elements [17, 27] As can be seen from Figure 2, block-like structures are observed when the underlying sequence shows homogeneous features over the duration of the corresponding segment. Often, such a homogeneous segment is followed by another homogeneous segment that stands in contrast to the previous one.

To identify the boundary between two homogeneous but contrasting segments (2D corner points), we slide a checkerboard kernel $\mathbf{K}$ along the main diagonal of the SSM and sum up the element-wise product of $\mathbf{K}$ and $\mathbf{S}$. A checkerboard kernel can be simply defined as a box kernel $\mathbf{K}_B \in \mathbb{Z}^{M \times M}$ where $M = 2L + 1$ is the size of the kernel, defined by $\mathbf{K}_B = sgn(k) \cdot sgn(l)\ \forall k, l \in [-L, L]$, where sgn is the sign function. This yields a novelty function $\Delta_{kernel}(n)$ for each index $n \in [1 : d]$ of $\mathbf{X}$ as follows:

$$\Delta_{kernel}(n) = \sum_{k,l \in [-L,L]} \mathbf{K}(k, l) \cdot \mathbf{S}(n + k, n + l) \tag{2}$$

for $n \in [L + 1 : d - L]$. When $\mathbf{K}$ is located within a relatively uniform region of S, the positive and negative values of the product tend to sum to zero (small novelty). Conversely, when $\mathbf{K}$ is at the crux of a checkerboard-like structure of S, the values of the product are all positive and sum up to a large value (high novelty) [50].

Local maxima of the novelty curve are then used to detect the boundaries of neighbouring segments that correspond to contrasting harmonic parts. For this, we use a pick peaking method that applies a smoothing filter to the novelty function (to reduce the effect of noise-like fluctuations) and uses adaptive thresholding to select a peak when novelty exceeds a local average [51]. The detected segment boundaries are used to split $\mathbf{X}$ and the corresponding $\mathbf{q}$ into a number of non-overlapping harmonic structures. This yields $\bar{\mathbf{q}} = \bar{\mathbf{q}}^1, \ldots, \bar{\mathbf{q}}^P$, where P denotes the number of structures.

3.3 Linking harmonic segments via similarity

Each harmonic structure $\bar{\mathbf{q}}^i$ is then considered for harmonic similarity. Since $\bar{\mathbf{q}}^i$ is still a time series (a partition of $\mathbf{q}$, the Harmonic Profile), we formulate the harmonic similarity between two harmonic structures by comparing their time series. This is done using Dynamic Time Warping (DTW) – an algorithm for comparing time series, which has been widely used across various domains, including speech recognition [49], pattern recognition [63], and bioinformatics [71]. In our case, DTW has desirable properties, as it is invariant to time shifts, and robust to local variations.

Vanilla DTW compares two time series by calculating the cumulative distances between each point/observation. It allows for non-linear alignment between the time series by considering the local warping path. The cost matrix, holding the cumulative distance between each corresponding point, is constructed using the Euclidean distance, and is formalised as:

$$d_{DTW}(\bar{\mathbf{q}}^i, \bar{\mathbf{p}}^j) = \sqrt{\sum_{(v,w) \epsilon \pi} ||\bar{q}^i_v + \bar{p}^i_w||^2} \tag{3}$$

where π is the optimal warping path – the shortest cumulative distance between the time series (found via dynamic programming).

As the computational complexity of vanilla DTW is quadratic in the sequence length, here we use the *Sakoe-Chiba* variant. The latter achieves linear complexity $O(N \cdot w)$, by constraining the warping path within a window of size w, rather than using all points (N).

Prior to the computation of similarities, time series are normalised are resampled to meet the same length, and standardised to zero mean and unit variance. This has the effect of comparing time series by looking at their shapes in an amplitude-invariant manner – which brings us closer to the identification of harmonic patterns.

The latter emerge after retrieving the k most similar structures for each segment $\bar{\mathbf{q}}^i$, and applying a filtration to retain only those

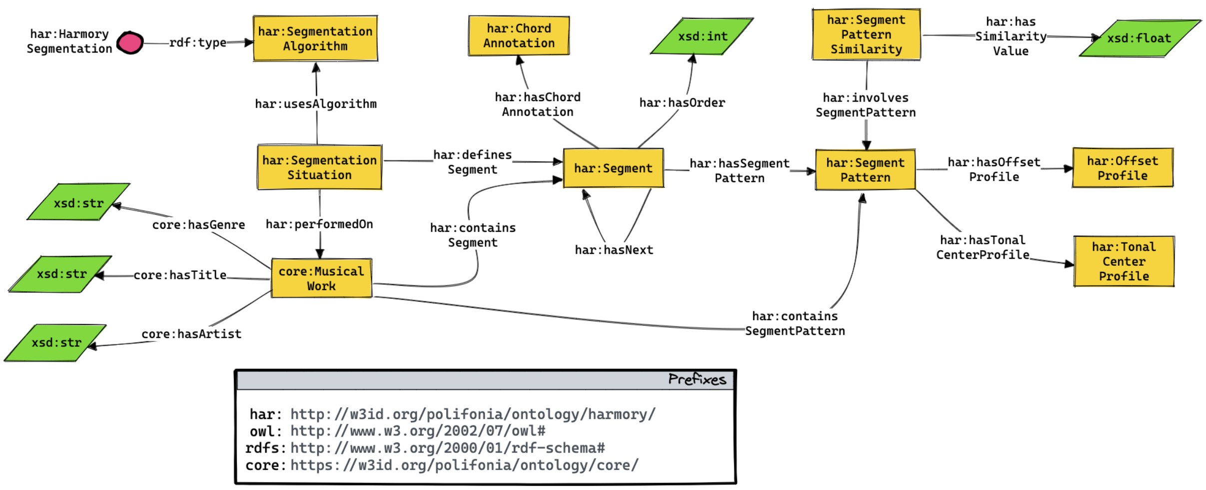

Figure 3: Illustration of the *Harmory Ontology*. The diagram uses the Grafoo notation, where yellow boxes are classes, blue/green arrows are object/datatype properties, green polygons are datatypes, and purple circles are individuals.

structures $\bar{\mathbf{p}}^i$ whose $d_{DTW}(\bar{\mathbf{q}}^i, \bar{\mathbf{p}}^j)$ is below a given threshold. Structures sharing the same (normalised) TPS time series ($d_{DTW} = 0$) define a distinct harmonic pattern; whereas segments with similar time series can be grouped within the same pattern family/cluster.

3.4 Knowledge graph creation

An ontology, called *Harmory Ontology*, was developed for the creation of the Knowledge Graph (KG). The ontology re-uses the Core module from the *Polifonia Ontology Network (PON)* [10], a network of ontologies for the semantic annotation of musical content. This allows to link Harmory to ChoCo[3] [16]. We also align to the *Music Ontology* [59] – a widely used ontology model in the music domain. The ontology was realised by applying the eXtreme Design methodology [56], which relies on Ontology Design Patterns (ODP) and on the iterative testing of the produced model against a predefined set of Competency Questions (CQs).

For each piece, the ontology allows to: (i) store its metadata, such as title, genre, and artist; (ii) hold the harmonic segmentation (see Section 3.2); and (iii) relate similar segments (see Section 3.3). This enables semantic access to the aforementioned data via SPARQL.

The model is illustrated in Figure 3, using the Grafoo notation[4]. A piece of music is expressed by means of the class `har:MusicalWork`, which is aligned with `mo:MusicalWork` in the Music Ontology. The metadata of a work is stored via `core:hasTitle`, `core:hasGenre`, and `core:hasArtist`, which describe the title, musical genre and composer or performer of the piece, respectively.

A musical work has a `har:SegmentationSituation` – a specialisation of the *Situation Pattern* [28] describing a segmentation performed by a specific `har:SegmentationAlgorithm` that produces one or more `has:Segments`. In this context, a harmonic sequence is split/partitioned into a number of segments, with their ordering allowing for reconstruction. Each sequence also holds its chords, using the class `mf:Chord`. Each segment is linked to its corresponding `har:SegmentPattern` – an abstraction of the TPS pattern normalised on the temporal axis. Hence, several `har:Patterns` may have the same `har:SegmentPattern`. Similarity relations are expressed via the class `har:SegmentPatternSimilarity`, which relates two Segment Patterns and holds their similarity value via the datatype property `har:hasSimilarityValue`.

4 EXPERIMENTS

To validate Harmory, we tested the efficacy of the two central components underpinning its creation: the DTW harmonic similarity (Section 3.3), and the harmonic segmentation (Section 3.2).

4.1 Evaluation of harmonic similarity

We evaluated the DTW harmonic similarity by comparing our implementation with other algorithms for the *cover song detection* task – a common benchmark for similarity algorithms in the symbolic music domain [18, 19].

In this comparison, performance is evaluated using two standard metrics: *First Tier* and *Second Tier*. The former measures the ratio of correctly retrieved songs within the top $(C_t - 1)$ matches to $(C_t - 1)$, where C_t is the size of the song class (e.g. the same composition, or performance) for track t. The First Tier can be formalised as:

$$FirstTier_{(D)} = \frac{1}{N} \sum_{t=0}^{N} \frac{||L_{|(C_t-1)|} \cap C_t||}{||(C_t - 1)||}, \tag{4}$$

where N is the set of all tracks in the dataset having at least a "cover", and $L_{(C_t-1)}$ denotes the list of matches for track t ranked by similarity – where only the first $(C_t - 1)$ occurrences are considered. Similarly, the Second Tier is defined as the ratio of correctly retrieved songs within the best $(2C_t - 1)$ matches to $(C_t - 1)$.

$$SecondTier_{(D)} = \frac{1}{N} \sum_{t=0}^{N} \frac{||L_{|(2C_t-1)|} \cap C_t||}{||(C_t - 1)||} \tag{5}$$

[3] ChoCo SPARQL endpoint: https://polifonia.disi.unibo.it/choco/sparql
[4] Graffo Notation: https://essepuntato.it/graffoo/

					Schubert		*CASD*		*Schubert+CASD*	
Algorithm	**TPS Mode**	**Stretch**	**Constraint**	**Normalise**	**First Tier**	**Second Tier**	**First Tier**	**Second Tier**	**First Tier**	**Second Tier**
TPSD	offset	-	-	-	0.49	0.63	0.62	0.68	0.58	0.67
TPSD	profile	-	-	-	0.53	0.74	0.76	0.83	0.69	0.8
DTW	offset	stretch	-	-	0.94	0.98	0.53	0.67	0.66	0.76
DTW	profile	stretch	-	-	0.97	0.99	0.6	0.69	0.71	0.78
DTW	offset	stretch	sakoe_chiba	-	0.96	0.99	0.62	0.7	0.72	0.79
DTW	profile	stretch	sakoe_chiba	-	0.97	0.99	0.69	0.77	0.77	**0.84**
DTW	offset	stretch	itakura	-	0.96	0.99	0.55	0.65	0.68	0.75
DTW	profile	stretch	sakoe_chiba	yes	0.97	0.99	0.7	0.76	**0.79**	0.83
LCSS	offset	-	sakoe_chiba	-	0.38	0.61	0.03	0.07	0.14	0.24
LCSS	offset	-	itakura	-	0.7	0.8	0.14	0.23	0.31	0.41
SoftDTW	offset	stretch	-	-	0.93	0.97	0.55	0.69	0.67	0.77
SoftDTW	profile	stretch	sakoe_chiba	-	0.98	0.99	0.62	0.73	0.73	0.81

Table 1: Performance of similarity algorithms on cover song detection. The highlighted lines denote the best performing algorithms, while results in bold indicate the best performance obtained for the *First Tier* and *Second Tier*, respectively.

Methods. We compare our DTW similarity (c.f. Section 3.3) with the following algorithms for harmonic and time series similarity:

- **Tonal Pitch Step Distance (TPSD)** [18, 19], a state of the art method that measures the difference between the *Harmonic Profiles* (see $\mathbf{q}$ in Section 3.2) of the given harmonic sequences. The difference is determined as the minimal area between the two time series, after considering all possible horizontal shifts. TPSD can handle sequences of different length, and has a time complexity of $O(nm \log(n+m))$, where n and m denote the length of the compared chord sequences [3];
- **Longest Common Subsequence (LCSS)** [68], a method expressing time series similarity based on their longest common subsequence. Similarity is calculated as the relative length of the longest common subsequence compared to the length of the shortest time series, thus ranging in $[0, 1]$. Using dynamic programming, LCSS is bounded in $O(n^2)$;
- **Soft Dynamic Time Warping (Soft DTW)** [15], a variant of DTW that allows for non-binary (fuzzy) alignments between time series, by using a sof-constraint. Soft DTW can be computed with quadratic time/space complexity.

All experiments are performed on the Harmonic Profile, in addition to an alternative formulation of the TPS time series, called *offsets*, where $q_i = \texttt{tps}(x_i, x_{i-1})$ (chord offset distance).

For *DTW*, *LCSS* and *Soft DTW*, two types of constraints were also tested: *Sakoe-Chiba* and *Itakura*. Analogously to Sakoe-Chiba, the Itakura constraint sets a maximum distance for each point in the time series, making the algorithm more efficient, and reducing the risk of being trapped in local minima. Several parameter settings for the Sakoe-Chiba radius and Itakura band were tested, and the best results were obtained by setting them to 5 and 4, respectively. This parametrisation turned out to be optimal across all our experiments.

Each method was tested on sequences of original length (*no-stretch*) and after resampling to the shortest sequence. We also experimented with normalised time series (Section 3.3).

Dataset. We two subsets of ChoCo [16] containing cover tracks: *Schubert Winterreise* [69] and *Chordify Annotator Subjectivity Dataset (CASD)* [38]. The former provides harmonic annotations for each of the 9 different performances of the same musical piece by Schubert. Similarly, CASD contains four annotations of the same performance, contributed by four different annotators. Chords from *Isophonics Dataset* [45] and *Jazz Audio-Aligned Harmony (JAAH)* [25] are also added to the reference dataset in order to add heterogeneity (different genres) and increase the complexity of the task.

Results. Table 1 shows the results of this comparison and highlights the best performing algorithms. Results are presented for Schubert and CASD separately, and also in a third merged setup (Schubert+CASD). Notably, the performance of the DTW algorithm is significantly better for Schubert (one piece, multiple performances), while for CASD (one performance, multiple annotations), TPSD performs slightly better. The best results for the third setup are obtained using the Sakoe-Chiba DTW, using normalisation and resampling on the shortest sequence. It is also worth remarking that our implementation, besides being the most accurate overall, is also the most efficient approach, due to its linear complexity.

4.2 Structural coverage of known patterns

To validate our harmonic segmentation (Section 3.2), we measure the overlap between the resulting structures with a collection of well-known chordal patterns. This exemplifies the hypothesis that a good segmentation would maximise the "reuse" of harmonic patterns – as building blocks that can be found in other pieces.

Given a segmentation $\bar{\mathbf{q}} = \bar{\mathbf{q}}^1, \ldots, \bar{\mathbf{q}}^P$ of a piece, with each $\bar{\mathbf{q}}^i$ containing a TPS time series, the overlap of $\bar{\mathbf{q}}$ with a dataset of known harmonic patterns $\mathcal{P}$ is computed as:

$$o_M(\bar{\mathbf{q}}) = \frac{1}{T} \sum_{i=1}^{T} \min_{\mathbf{p} \in \mathcal{P}} d_{\mathsf{DTW}}(\bar{\mathbf{q}}^i, \mathbf{p}), \tag{6}$$

$$o_B(\bar{\mathbf{q}}) = \min_{\bar{\mathbf{q}}^i \in \bar{\mathbf{q}}} \min_{\mathbf{p} \in \mathcal{P}} d_{\mathsf{DTW}}(\bar{\mathbf{q}}^i, \mathbf{p}), \tag{7}$$

which differ in the aggregation function. The former measures the average pattern distance contributed by each structure in the segmentation; the latter, instead, only retains the distance of the most similar pattern that was matched to one of the structures. When $o_M = 0$, all segments are fully matched/found in $\mathcal{P}$; whereas o_M is minimal when at least a segment matches a pattern in $\mathcal{P}$.

Methods. We compare our method (denoted as *harmov*) to fast low-cost unipotent semantic segmentation (FLUSS) [29] – a state of the art algorithm for time series segmentation defined on the Matrix

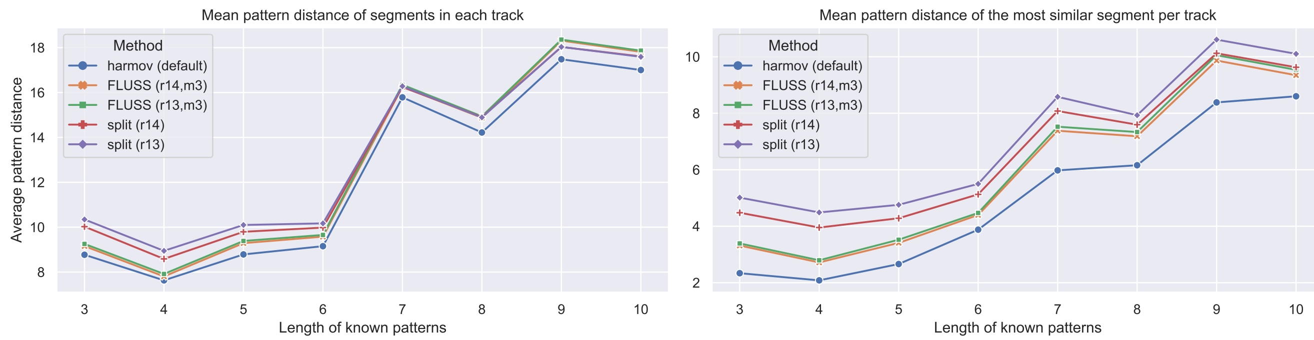

Figure 4: Structural coverage of known patterns for each segmentation method, using Equation 4 (*left*), and Equation 5 (*right*). Results are reported as distances averaged per pattern group (a group contains known harmonic patterns of the same length).

Profile [70]. FLUSS annotates the time series with information about the likelihood of a regime change (a segment boundary); and is parameterised by a fixed window size m, and the number of segments to detect r. We also include a baseline splitting a time series in r uniform segments. Both methods operates on the TPS Profile of $\mathbf{h}$, and are optimised via grid search.

Datasets. We compute and evaluate the harmonic segmentations on a dataset comprising 320 chord progressions, obtained from randomly sampling 40 pieces per audio partition in ChoCo (`isophonics`, `billboard`, `casd`, `schubert-winterreise`, `rwc-pop`, `uspop-2002`, `jaah`, `robbie-williams`). This yields a diversified (several genres, durations, etc.) yet representative sample of Harmory (≈ 2% of ChoCo); which prevents larger partitions from biasing the overall results. For $\mathcal{P}$, we assembled a dataset of known harmonic patterns from Impro-Visor [57], which is available on GitHub[5]. After filtration of trivial occurrences (e.g. chord uni-grams, sequences with repeated chord occurrences, etc.), the dataset counts 300 unique patterns spanning from 3 to 10 chord occurrences per pattern (the length of a chordal pattern).

Results. The structural coverage, computed for each segmentation method and aggregated for all known harmonic patterns of the same length, is reported in Figure 4. For both measures o_M, o_B, the segmentations produced by our method (*harmov*) produce the lowest distances – meaning that they show the highest overlap with the known harmonic patterns in $\mathcal{P}$. This behaviour is preserved for all pattern groups (the x-axis), and the gap with the other methods increases with pattern's length. The second performing method is FLUSS, using $r = 14$ split regions and a window size of $m = 3$. However, for longer patterns, the latter performs comparably with a fixed sequence split (the other baseline). Finally, it is worth remarking that results for all baselines are first optimised on a grid search; whereas we use the default parameterisation for *harmov*.

5 AVENUES FOR MACHINE CREATIVITY

We envisage various applications of Harmory across different tasks and use cases, ranging from music information retrieval and computational musicology, to creativity support for artistic workflows. The latter is the main focus of this work. However, we do not aim at improving the state of the art in music generation, but rather to provide a transparent system to support creative workflows [11].

Here, we show examples of trustworthy applications for pattern discovery, human-machine chord generation, and harmonic similarity. The latter is more of musicological interest, whereas the former are both creative use cases. Each application is demonstrated through a set of *Prompts*, expressed in natural language, which correspond to SPARQL query templates to interrogate the KG (Section 3). The latter are fully available on our repository[6].

5.1 Pattern discovery

The traversal of the Harmonic Memory makes it possible to obtain granular information of the harmonic structure of songs. In particular, it possible to explore the harmonic segments of each song, the patterns related to each segment, and the similarities with other patterns/segments found in other pieces.

A composer may start with a harmonic pattern mind, and initiate a creative exploration of the KG by leveraging music and metadata.

Prompt 1 *For a given pattern, which are the tracks (titles, artists and genres) in which the pattern can be found?*

Prompt 2 *Given a music genre, what are the most frequent patterns?*

To support creative exploration, more complex prompts can be formulated in order to narrow down the search, and eventually discover surprising or unexpected outputs, if present.

Prompt 3 *Which harmonic patterns are used in "Michelle" by The Beatles, but also in a classical composition?*

Prompt 4 *Which patterns used by The Beatles in "Michelle" but not in "Hey Jude" contain at least a B flat major seventh chord?*

In the Harmory KG, we have included known patterns (as described in Section 4.2), which are labelled in such a way as to indicate their origin, mood, or harmonic function within the progression. These labelled harmonic fragments can be used as input for a query, e.g. for searching songs that contain them:

Prompt 5 *Which tracks include a dominant cycle in seven steps?*

[5]https://github.com/Impro-Visor/Impro-Visor

[6]SPARQL queries: https://github.com/smashub/harmory/tree/main/queries

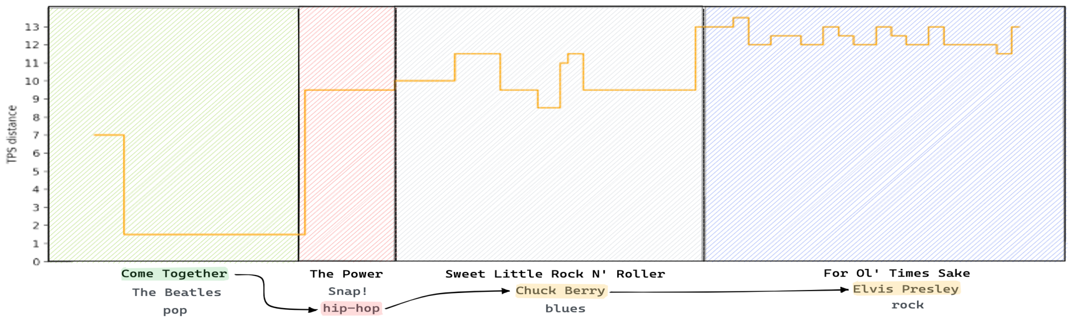

Figure 5: Example of a generated chord progression using a pattern-based prompt. Given a first segment, each segment is chosen according to similarity to the subsequent one in the original sequence, and filtered according to arbitrary criteria. The second segment is taken from a song that has *"hip-hop"* as genre, while the next two segments are chosen by *artist*.

5.2 Human-machine chord generation

Harmory also enables combinational creativity use cases. New progressions are generated by moving across patterns through temporal and similarity links, based on the given creative requirements. At generation time, this has the advantage of giving recognition to all artists that contributed to the new creation, as shown in Figure 5.

First, it can provide statistical information regarding variations of a given harmonic sequence. As these variations come from real pieces, it is also possible to leverage metadata for controlling the generation. To do this, a prompt can be formulated from a given (possibly new) harmonic sequence (or a part of it), to retrieve all the all harmonic sequences using the same pattern.

Prompt 6 *Given a chord sequence, which are its variations, and which tracks these variations belong to?*

Similarly, it is also possible to query the most similar (or most distant) harmonic sequences to a given one:

Prompt 7 *Given a chord sequence, which are its most similar chord sequences, sorted by similarity?*

These simple constructs already allow to generate new harmonic sequences, starting from either a known harmonic idea/pattern, or a full progression. If starting from a full progression, one way is to identify the first harmonic segment that makes it up. From this point, transitions can be made using *similarity relations*, while taking into account the order of the different segments (*temporal connections*) and their tonality. For example, starting from the first harmonic segment of a song (a priming sequence), one can then generate a continuation by identifying similar sequences to the next sequence, filtering them by tonality (or/and by artist, genre, title) and repeat this process recursively for a number of steps, criteria, or with the supervision/control of the user.

Prompt 8 *Create a progression starting with "Michelle" by The Beatles, continuing with a segment found in a classical piece of music, and then continuing with another by Chet Baker.*

5.3 Harmonic similarity

From a musicological perspective, the KG can also be used to analyse similarity relations between tracks – by leveraging the local information relating harmonic structures. This also allows for the formal definition of similarity functions (depending on a genre- or task- specific notions) by using logical operators (SPARQL syntax) over harmonic segments/patterns. An example is given below.

Prompt 9 *Given a track, which tracks contain patterns with a distance of less than* 0.2*, each having the same order?*

As expected, the results of this query are almost exclusively cover songs of the given track. Nevertheless, a similarity function can be defined to be less strict, and hence more explorative. For instance, the similarity function below uses a higher similarity threshold for patterns, and does not constrain on the order of segments.

Prompt 10 *Given a track, which tracks contain patterns with a distance of less than* 0.5*, regardless of their order?*

6 CONCLUSIONS

Our work contributes a Web resource aimed to support the design of trustworthy systems for computational music creativity. This is a central requirement for the large scale adoption of these systems, which is often neglected in generative machine learning research. To this vision, we leverage a corpus of harmonic annotations on the Web, to design the Harmonic Memory (Harmory) – a Knowledge Graph of interconnected chordal patterns which is perceptually and musicologically grounded. After demonstrating the validity of our framework, we showed how Harmony can enable transparent, explainable, and accountable applications for human-machine creativity – ranging from pattern discovery and chord generation, to harmonic similarity. Future work includes linking Web resources with musical pieces in Harmory, to achieve the alignment to other ontologies and Knowledge Graphs in the music domain. We also envisage the inclusion of heterogeneous data to enrich and complement the harmonic information, such as perceptual metadata, musical content (melodies), and complexity measures.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004746.

REFERENCES

[1] Eytan Agmon. 1995. Functional Harmony Revisited: A Prototype-Theoretic Approach. *Music Theory Spectrum* 17, 2 (10 1995), 196–214. https://doi.org/10.2307/745871 arXiv:https://academic.oup.com/mts/article-pdf/17/2/196/6138625/17-2-196.pdf

[2] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse H. Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. 2023. MusicLM: Generating Music From Text. *CoRR* abs/2301.11325 (2023). https://doi.org/10.48550/arXiv.2301.11325

[3] Greg Aloupis, Thomas Fevens, Stefan Langerman, Tomomi Matsui, Antonio Mesa, Yurai Núñez Rodríguez, David Rappaport, and Godfried T. Toussaint. 2006. Algorithms for Computing Geometric Measures of Melodic Similarity. *Comput. Music. J.* 30, 3 (2006), 67–76. https://doi.org/10.1162/comj.2006.30.3.67

[4] Margaret A Boden. 1992. Understanding creativity. *The Journal of Creative Behavior* (1992).

[5] Margaret A Boden. 2004. *The creative mind: Myths and mechanisms.* Routledge.

[6] Paul M. Bodily and Dan Ventura. 2018. Explainability: An Aesthetic for Aesthetics in Computational Creative Systems. In *Proceedings of the Ninth International Conference on Computational Creativity, ICCC 2018, Salamanca, Spain, June 25-29, 2018*, François Pachet, Anna Jordanous, and Carlos León (Eds.). Association for Computational Creativity (ACC), 153–160. http://computationalcreativity.net/iccc2018/sites/default/files/papers/ICCC_2018_paper_42.pdf

[7] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. 2020. *Deep learning techniques for music generation.* Vol. 1. Springer.

[8] Nick Bryan-Kinns, Berker Banar, Corey Ford, C Reed, Yixiao Zhang, Simon Colton, Jack Armitage, et al. 2022. Exploring xai for the arts: Explaining latent space in generative music. (2022).

[9] Benjamin Burger, Phillip M Maffettone, Vladimir V Gusev, Catherine M Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M Alston, Buyi Li, Rob Clowes, et al. 2020. A mobile robotic chemist. *Nature* 583, 7815 (2020), 237–241.

[10] Valentina Anita Carriero, Fiorela Ciroku, Jacopo de Berardinis, Delfina Sol Martinez Pandiani, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. 2021. Semantic Integration of MIR Datasets with the Polifonia Ontology Network. In *ISMIR Late Breaking Demo.*

[11] Shan Carter and Michael Nielsen. 2017. Using artificial intelligence to augment human intelligence. *Distill* 2, 12 (2017), e9.

[12] Nick Collins, V Ruzicka, and Mick Grierson. 2020. Remixing AIs: mind swaps, hybrainity, and splicing musical models. In *Proc. The Joint Conference on AI Music Creativity.*

[13] Simon Colton, John William Charnley, and Alison Pease. 2011. Computational Creativity Theory: The FACE and IDEA Descriptive Models. In *ICCC*. Mexico City, 90–95.

[14] Simon Colton and Dan Ventura. 2014. You Can't Know my Mind: A Festival of Computational Creativity. In *Proceedings of the Fifth International Conference on Computational Creativity, ICCC 2014, Ljubljana, Slovenia, June 10-13, 2014*, Simon Colton, Dan Ventura, Nada Lavrac, and Michael Cook (Eds.). computationalcreativity.net, 351–354. http://computationalcreativity.net/iccc2014/wp-content/uploads/2014/06/15.8_Colton.pdf

[15] Marco Cuturi and Mathieu Blondel. 2017. Soft-DTW: a Differentiable Loss Function for Time-Series. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 894–903. https://proceedings.mlr.press/v70/cuturi17a.html

[16] Jacopo de Berardinis, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. 2023. ChoCo: a Chord Corpus and a Data Transformation Workflow for Musical Harmony Knowledge Graphs. In *Manuscript under review.*

[17] Jacopo de Berardinis, Michalis Vamvakaris, Angelo Cangelosi, and Eduardo Coutinho. 2020. Unveiling the hierarchical structure of music by multi-resolution community detection. *Transactions of the International Society for Music Information Retrieval* 3, 1 (2020), 82–97.

[18] W. Bas de Haas, Remco C. Veltkamp, and Frans Wiering. 2008. Tonal Pitch Step Distance: a Similarity Measure for Chord Progressions. In *ISMIR 2008, 9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, PA, USA, September 14-18, 2008*, Juan Pablo Bello, Elaine Chew, and Douglas Turnbull (Eds.). 51–56. http://ismir2008.ismir.net/papers/ISMIR2008_252.pdf

[19] W. Bas de Haas, Frans Wiering, and Remco C. Veltkamp. 2013. A geometrical distance measure for determining the similarity of musical harmony. *International Journal of Multimedia Information Retrieval* 2, 3 (01 09 2013), 189–202. https://doi.org/10.1007/s13735-013-0036-6

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

[21] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341* (2020).

[22] Eric Drott. 2021. Copyright, compensation, and commons in the music AI industry. *Creative Industries Journal* 14, 2 (2021), 190–207.

[23] M. du Sautoy. 2019. *The Creativity Code: How AI is learning to write, paint and think.* HarperCollins Publishers.

[24] Wlodzislaw Duch. 2007. Intuition, insight, imagination and creativity. *IEEE Computational Intelligence Magazine* 2, 3 (2007), 40–52.

[25] Vsevolod Eremenko, Emir Demirel, Baris Bozkurt, and Xavier Serra. 2018. *JAAH: Audio-aligned jazz harmony dataset.* https://doi.org/10.5281/zenodo.1290737

[26] Luciano Floridi. 2019. Establishing the rules for building trustworthy AI. *Nature Machine Intelligence* 1, 6 (2019), 261–262.

[27] Jonathan Foote. 1999. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1).* 77–80.

[28] Aldo Gangemi and Peter Mika. 2003. Understanding the Semantic Web through Descriptions and Situations. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Robert Meersman, Zahir Tari, and Douglas C. Schmidt (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 689–706.

[29] Shaghayegh Gharghabi, Yifei Ding, Chin-Chia Michael Yeh, Kaveh Kamgar, Liudmila Ulanova, and Eamonn Keogh. 2017. Matrix profile VIII: domain agnostic online semantic segmentation at superhuman performance levels. In *2017 IEEE international conference on data mining (ICDM).* IEEE, 117–126.

[30] Zixun Guo, Jaeyong Kang, and Dorien Herremans. 2022. A Domain-Knowledge-Inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling. *arXiv preprint arXiv:2212.00973* (2022).

[31] Christopher Harte, Mark B Sandler, Samer A Abdallah, and Emilia Gómez. 2005. Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations.. In *ISMIR*, Vol. 5. 66–71.

[32] Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron C. Courville, and Douglas Eck. 2017. Counterpoint by Convolution. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, Sally Jo Cunningham, Zhiyao Duan, Xiao Hu, and Douglas Turnbull (Eds.). 211–218.

[33] Cheng-Zhi Anna Huang, Hendrik Vincent Koops, Ed Newton-Rex, Monica Dinculescu, and Carrie J Cai. 2020. AI song contest: Human-AI co-creation in songwriting. *arXiv preprint arXiv:2010.05388* (2020).

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

[35] Jaehun Kim, Julián Urbano, Cynthia Liem, and Alan Hanjalic. 2020. One Deep Music Representation to Rule Them All?: A Comparative Analysis of Different Representation Learning Strategies. *Neural Computing and Applications* 32, 4 (2020), 1067–1093.

[36] Arto Klami, Theodoros Damoulas, Ola Engkvist, Patrick Rinke, and Samuel Kaski. 2022. Virtual Laboratories: Transforming research with AI. (2022).

[37] Stefan Koelsch. 2011. Toward a neural basis of music perception–a review and updated model. *Frontier in Psychology* 2 (2011), 110.

[38] Hendrik Vincent Koops, W. Bas de Haas, John Ashley Burgoyne, Jeroen Bransen, Anna Kent-Muller, and Anja Volk. 2019. Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research* 48, 3 (2019), 232–252. https://doi.org/10.1080/09298215.2019.1613436

[39] Allison Lahnala, Gauri Kambhatla, Jiajun Peng, Matthew Whitehead, Gillian Minnehan, Eric Guldan, Jonathan K. Kummerfeld, Anil Çamci, 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 - 10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual Event, April 7-9, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12693)*, Juan Romero, Tiago Martins, and Nereida Rodríguez-Fernández (Eds.). Springer, 171–186. https://doi.org/10.1007/978-3-030-72914-1_12

[40] Agnieszka Lawrynowicz. 2020. Creative AI: A new avenue for the Semantic Web? *Semantic Web* 11 (2020), 69–78. https://doi.org/10.3233/SW-190377 1.

[41] Nicolas Lazzari, Andrea Poltronieri, and Valentina Presutti. 2022. Pitchclass2vec: Symbolic Music Structure Segmentation with Chord Embeddings. In *Proceedings of the 1st Workshop on Artificial Intelligence and Creativity co-located with 21th International Conference of the Italian Association for Artificial Intelligence(AIxIA 2022), Udine, Italy, November 28 - December 3, 2022 (CEUR Workshop Proceedings, Vol. 3278)*, Allegra De Filippo, Michela Milano, Valentina Presutti, and Alessandro Saffiotti (Eds.). CEUR-WS.org, 14–30. http://ceur-ws.org/Vol-3278/paper2.pdf

[42] Fred Lerdahl. 1988. Tonal Pitch Space. *Music Perception: An Interdisciplinary Journal* 5, 3 (1988), 315–349. http://www.jstor.org/stable/40285402

[43] Fred Lerdahl and Ray Jackendoff. 1983. *A generative theory of tonal music.* The MIT Press, Cambridge. MA.

[44] Maria Teresa Llano, Mark d'Inverno, Matthew Yee-King, Jon McCormack, Alon Ilsar, Alison Pease, and Simon Colton. 2020. Explainable Computational Creativity. In *Proceedings of the Eleventh International Conference on Computational Creativity, ICCC 2020, Coimbra, Portugal, September 7-11, 2020*, F. Amílcar Cardoso, Penousal Machado, Tony Veale, and João Miguel Cunha (Eds.). Association for Computational Creativity (ACC), 334–341. http://computationalcreativity.net/

iccc20/papers/067-iccc20.pdf

[45] Matthias Mauch, Simon Dixon, Christopher Harte, et al. 2007. Discovering chord idioms through Beatles and Real Book songs.

[46] Rick Meerwaldt, Albert Meroño-Peñuela, and Stefan Schlobach. 2017. Mixing Music as Linked Data: SPARQL-based MIDI Mashups. In *WHiSe@ ISWC*. 87–98.

[47] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1301.3781

[48] Fabio Morreale. 2021. Where Does the Buck Stop? Ethical and Political Issues with AI in Music Creation. *Transactions of the International Society for Music Information Retrieval* (Jul 2021).

[49] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *CoRR* abs/1003.4083 (2010). http://dblp.uni-trier.de/db/journals/corr/corr1003.html#abs-1003-4083

[50] Meinard Müller. 2021. *Fundamentals of music processing: Using Python and Jupyter notebooks*. Vol. 2. Springer.

[51] Oriol Nieto and Juan Pablo Bello. 2016. Systematic Exploration of Computational Music Structure Research. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*. ISMIR, New York City, United States, 547–553. https://doi.org/10.5281/zenodo.1417661

[52] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155* (2022).

[53] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://doi.org/10.3115/v1/d14-1162

[54] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[55] Azzurra Pini, Jer Hayes, Connor Upton, and Medb Corcoran. 2019. AI Inspired Recipes: Designing Computationally Creative Food Combos. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312948

[56] Valentina Presutti, Enrico Daga, Aldo Gangemi, and Eva Blomqvist. 2009. eXtreme Design with Content Ontology Design Patterns. In *Proceedings of the Workshop on Ontology Patterns (WOP 2009) , collocated with the 8th International Semantic Web Conference (ISWC-2009), Washington D.C., USA, 25 October, 2009 (CEUR Workshop Proceedings, Vol. 516)*, Eva Blomqvist, Kurt Sandkuhl, François Scharffe, and Vojtech Svátek (Eds.). CEUR-WS.org. http://ceur-ws.org/Vol-516/pap21.pdf

[57] Alexander M Putman and Robert M Keller. 2015. A transformational grammar framework for improvisation. In *First International Conference on New Music Concepts*.

[58] 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 *International Conference on Machine Learning*. PMLR, 8821–8831.

[59] Yves Raymond, Samer Abdallah, Mark Sandler, and Frederick Giasson. 2007. The Music Ontology. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*. Vienna, Austria, 417–422.

[60] 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*. PMLR, 4364–4373.

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[62] Amaia Salvador, Michal Drozdzal, Xavier Giró-i Nieto, and Adriana Romero. 2019. Inverse cooking: Recipe generation from food images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10453–10462.

[63] KC Santosh, Bart Lamiroy, and Laurent Wendling. 2013. DTW–Radon-based shape descriptor for pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence* 27, 03 (2013), 1350008.

[64] Bob Sturm, Joao Felipe Santos, and Iryna Korshunova. 2015. Folk music style modelling by recurrent neural networks with long short term memory units. In *16th international society for music information retrieval conference*.

[65] Bob LT Sturm, Maria Iglesias, Oded Ben-Tal, Marius Miron, and Emilia Gómez. 2019. Artificial intelligence and music: open questions of copyright law and engineering praxis. In *Arts*, Vol. 8. MDPI, 115.

[66] Iiris Sundin, Alexey Voronov, Haoping Xiao, Kostas Papadopoulos, Esben Jannik Bjerrum, Markus Heinonen, Atanas Patronov, Samuel Kaski, and Ola Engkvist. 2022. Human-in-the-loop assisted de novo molecular design. *Journal of Cheminformatics* 14, 1 (2022), 1–16.

[67] Peter M Todd and Gregory M Werner. 1999. Frankensteinian methods for evolutionary music composition. *Musical networks: Parallel distributed perception and performance* 3, 4 (1999), 7.

[68] M. Vlachos, G. Kollios, and D. Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings 18th International Conference on Data Engineering*. 673–684. https://doi.org/10.1109/ICDE.2002.994784

[69] Christof Weiß, Frank Zalkow, Vlora Arifi-Müller, Meinard Müller, Hendrik Vincent Koops, Anja Volk, and Harald G Grohganz. 2021. Schubert Winterreise dataset: A multimodal scenario for music analysis. *Journal on Computing and Cultural Heritage (JOCCH)* 14, 2 (2021), 1–18.

[70] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee, 1317–1322.

[71] Yuan Yuan, Yi-Ping Phoebe Chen, Shengyu Ni, Augix Guohua Xu, Lin Tang, Martin Vingron, Mehmet Somel, and Philipp Khaitovich. 2011. Development and application of a modified dynamic time warping algorithm (DTW-S) to analyses of primate brain expression time series. *BMC bioinformatics* 12 (2011), 1–13.