THE JAZZ HARMONY TREEBANK

Daniel Harasim[1] **Christoph Finkensiep**[1] **Petter Ericson**[1]
Timothy J. O'Donnell[2] **Martin Rohrmeier**[1]
[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland
[2] Department of Linguistics, McGill University, Canada
daniel.harasim@epfl.ch

ABSTRACT

Grammatical models which represent the hierarchical structure of chord sequences have proven very useful in recent analyses of Jazz harmony. A critical resource for building and evaluating such models is a ground-truth database of syntax trees that encode hierarchical analyses of chord sequences. In this paper, we introduce the Jazz Harmony Treebank (JHT), a dataset of hierarchical analyses of complete Jazz standards. The analyses were created and checked by experts, based on lead sheets from the open iRealPro collection. The JHT is publicly available in JavaScript Object Notation (JSON), a human-understandable and machine-readable format for structured data. We additionally discuss statistical properties of the corpus and present a simple open-source web application for the graphical creation and editing of trees which was developed during the creation of the dataset.

1. INTRODUCTION

Jazz music exhibits hierarchical relations between chords. This is particularly apparent in the fact that virtually any chord of a Jazz standard can be prepared by an applied dominant or subdominant. In fact, many chord sequences can be explained as the recursive application of such preparations [41]. Chords that are far apart in time can therefore be directly related, establishing long-range dependencies that can span whole formal sections of pieces. Such hierarchical structures also correlate with empirical findings from music perception research [25]. This is by no means to say that hierarchies are the only relevant relations between chords. Hierarchical chord relations are, however, underrepresented in computational models of harmony to date; the here presented dataset is intended to ease the development of hierarchical models.

Inspired by Schenkerian theory [3, 45] and generative syntax formalisms for natural language, generative theories of harmonic syntax model the hierarchical relations in chord sequences based on formal grammatical devices such as context-free grammars. Recent research uses formal grammars to represent hierarchical relations in melodies [1,10,13,16,24,34], chord sequences [15,19,43], and rhythms [18,29]. The fields of application include music theory [37, 40], music psychology [25, 42], automatic harmonic analysis [7, 8], and automatic music transcription [11,30,35].

The aim of this article is to present the Jazz Harmony Treebank (JHT), a dataset of hierarchical harmonic analyses of Jazz standards by music experts in a human-understandable and machine-readable format. We report on the creation of the treebank, describe the musical interpretation of the syntax trees, and explain the decisions that were made to meet the challenges of the annotation procedure. The dataset is available on GitHub. [1]

Treebanks are of particular importance for the study of hierarchical models and their applications. In linguistics, they have been and remain instrumental for many natural language processing tasks. The well-known Penn Treebank [28], first published in the early nineties, is an instructive example since it has been used as an object of study in and of itself [12], as a basis for publishing additional treebanks with different paradigms [21] and for different languages [27], and–most prominently–as a dataset for training and evaluating machine-learning methods [22, 31, 44].

The present article describes the creation process of the JHT. We take this as an opportunity to study the details of harmonic syntax using several concrete examples of Jazz standards. The major challenge of this application lies in the many individual decisions analysts have to take to address the ambiguity of music. Importantly, our goal is not to create uniform syntax trees of Jazz chord sequences, but to describe individual and subjective listening experiences in an unambiguous formal representation. Harmonic relations in sufficiently long chord sequences can be perceived in several ways, without one interpretation being clearly preferable. Therefore, the syntax trees of the JHT are best understood as proposals with a clear interpretation. The trees provide a basis for further analytical discussions, sophisticated computational models, and for education.

1.1 Related Symbolic Datasets

Many existing collections of symbolic data about chord sequences concentrate on providing chord labels for harmonic entities. Two prominent datasets of time-aligned

[1] https://github.com/DCMLab/JazzHarmonyTreebank

chord symbols were created by Harte et al. [20] and Burgoyne et al. [2] to study automatic chord transcription from audio. Neuwirth et al. [36] take a more music-theoretically motivated approach by proposing a chord-symbol representation for Western classical music and apply it to scale degree analyses of Beethoven's string quartets. Chen and Su [5] and Devaney et al. [9] similarly label excerpts from common-practice tonality. Micchi et al. [32] combine existing Roman numeral analyses into a meta-dataset.

The datasets just mentioned use chord labels to analyze music given as audio data or in a symbolic representation. Since we analyze the relations between the chords of such sequences, this study is located at a higher level of abstraction. Only a few datasets of hierarchical analyses of sequential musical data are available in divergent formats [39]. Hamanaka et al. [17] and Kirlin [23] created two datasets of tree analyses of melodies of Western Classical Music. Gotham and Ireland [14] study musical form by the creation of datasets in a hierarchical representation. Moss et al. [33] study Brazilian Choro using a dataset with hierarchical form encoding. Granroth-Wilding and Steedman [15] provide a dataset of 76 sub-sequences of Jazz standards with partial harmonic grouping labels. In contrast to previous research that analyzed snippets of musical pieces, the JHT consists of 150 full chord sequences of Jazz standards with complete harmonic syntax trees.

2. HARMONIC SYNTAX

A harmonic syntax tree, as shown in Figure 1a, denotes a mental representation of a musical piece as a whole. Unlike sequential models that describe how, for instance, a sequence of chord symbols is generated chord by chord from the start to the end, hierarchical models describe how the skeleton of a piece is generated and recursively elaborated [43]. In Jazz, the most prominent of those elaboration operations are the duplication of chords and the preparation of a chord by an applied dominant. Each application of an operation establishes a direct relation between two chords. A syntax tree consists exactly of the sum of all those relations. It is therefore not directly a model for first-time listening of a musical piece, but rather for the abstract representation of musicians or listeners who are (implicitly or explicitly) aware of a piece's harmonic relations. This usage of the word *syntax* is closely related to generative syntax formalisms of natural language that address the question of which relations between words a listener must notice to understand the meaning of a sentence [6].

The scope of this paper is limited to tonal Jazz, including Swing, Bossa Nova, Jazz Blues, Bebop, Cool Jazz, and Hard Bop, and excluding parts of traditional Blues, Modal Jazz, Free Jazz, and Modern Jazz. We furthermore excluded tunes such as *Groovin' High* whose harmonic structure requires even more expressive representations than trees. [2] The general idea of harmonic syntax is, however, also applicable to other musical styles such as Western classical music.

2.1 Prolongation and Preparation as Fundamental Principles

In the following, we present the syntactic formalism with a particular emphasis on its musical interpretation. The concept of functional harmony describes an expectation-realization structure between musical objects such as notes, chords, and keys. Consider for example the chords of the final cadence of the Jazz standard *Birk's Works*, Fm6 Abm7 Db7 G%7 C7 Fm6, where G%7 denotes a half-diminished seventh chord with root G. Figure 1b shows the expectation-realization structure of this chord sequence. The first Fm6 establishes the tonic and as such creates the expectation that the progression ends with Fm6. The chords Abm7 and Db7 function as the tritone-substituted subdominant and dominant of C7, respectively. They therefore create expectation that resolves in the (temporally distant) chord C7. The chord G%7 functions as a subdominant chord in F minor. It therefore creates expectation that resolves with the dominant chord C7 which itself resolves into the last tonic chord Fm6. We say that the tonic chords constitute a *prolongation*. The subdominant chords *prepare* the dominant chords and the dominant chords *prepare* the tonic chord. Abstractly, we say that a chord X *refers* to a chord Y if X either prolongs or prepares Y. [3]

Prolongation and preparation are the two fundamental principles of functional harmonic syntax [41]. They can be formalized as rules of a context-free grammar with chord symbols both as terminals and nonterminals. In the formalization, *strong prolongations* that prolong chords of the same root and chord form are distinguished from *weak prolongations* that prolong a chord with a functionally equivalent chord (e.g., prolongation of C with Am). Note that this concept of weak prolongation is more general than in the GTTM where prolonging chords are for instance required to have the same root [26]. Strong prolongation is represented by rules of the form $X \longrightarrow X\ X$ for chord symbols X (e.g., Fm6 $\longrightarrow$ Fm6 Fm6). For chord symbols X and Y, rules of the form $X \longrightarrow Y\ X$ and $X \longrightarrow X\ Y$ represent weak prolongations if X and Y are functionally equivalent (e.g., Fm6 $\longrightarrow$ Ab Fm6). If otherwise X and Y are not functionally equivalent, $X \longrightarrow Y\ X$ represents a preparation (e.g., Fm6 $\longrightarrow$ C7 Fm6).

The practise of having no separate alphabet of nonterminal symbols, and requiring each binary rule to have a left-hand side symbol also on the right-hand side, is related to dependency grammars [38] and categorical grammars [47] which are well-known in computational linguistics and natural language processing. The symbol that appears both on the left-hand side and the right-hand side is called the *head* of the rule. In our setting of prolongation and preparation, the prolonged (resp. prepared) chord is the head. Therefore, weak prolongation rules may be left- or right-headed, while preparation rules are always right-headed. In sum, our harmony grammar consists of the following rules which model strong prolongation, weak pro-

[2] *Groovin' High* exhibits crossing harmonic dependencies between a tonic prolongation from m1 to m5 and a dominant preparation from m4 to m7. A similar tune is *Out of Nowhere*.

[3] In contrast to models based on the *Generative Theory of Tonal Music* [26], we exclude the concept of departure as a primitive relation, because it is not consistent with our formalization of the expectation-realization structure.

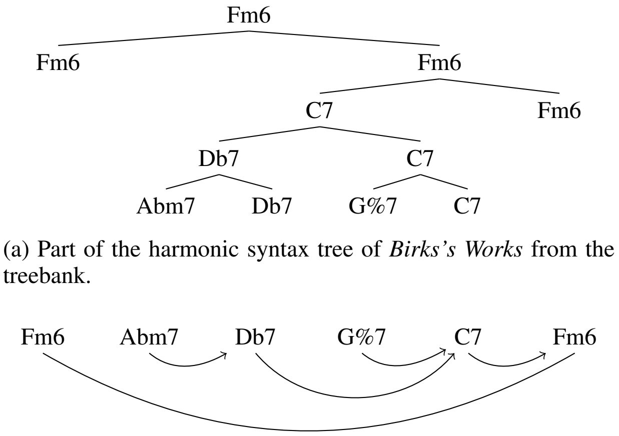
longation, and preparation, respectively,

$X \longrightarrow X\ X$	for any chord X	(strong prol.)
$X \longrightarrow Y\ X \mid X\ Y$	for any chord X and a functionally equivalent chord Y	(weak prol.)
$X \longrightarrow Y\ X$	for any chord X and a chord Y that prepares X	(preparation)

The tree in Figure 1a is a parse tree of the chord sequence Fm6 Abm7 Db7 G%7 C7 Fm6 under such a grammar of harmonic structure. Those parse trees represent exactly the same information as expectation-realization structures such as shown in Figure 1b: Undirected edges correspond to strong prolongations and directed edges correspond to either weak prolongations or preparations. This short example is unambiguous–it has only one plausible syntactic structure. In general, however, there are many syntax trees possible for a chord sequence. Grammar rules and syntax trees can then be weighted by probabilities that capture the plausibility of an analysis [1, 19, 24]. To identify the syntax tree that most accurately describes one's perception of the harmonic structure, other dimensions such as rhythm, form, and melody must also be taken into account. Even the artistic interpretation of a musical performance and the individual musical background of listeners have the potential to influence the perceived harmonic structure of a piece. A formal grammar that purely models chord symbols can therefore only answer the question "Is this a plausible syntax tree for a Jazz standard?", but not the question "Is this tree a good analysis of that particular tune in a particular context?". Until more complete models of musical structure are developed that integrate all relevant musical dimensions, the second question can only be answered by humans.

2.2 Complete Constituents and Open Constituents

Constituents formalize the notion of a musical unit such as a chord or a phrase. In the syntax tree shown in Figure 1a, the complete constituents are exactly the subsequences that are leafs of single subtrees, such as the subsequence Abm7 Db7 G%7 C7. Formally, we call a subsequence a *complete constituent* if it contains a chord, called the *head*, that is transitively referred to by all other chords of the sequence.[4] For instance, the chord C7 is the head of the phrase Abm7 Db7 G%7 C7 and Fm6 is the head of the whole sequence Fm6 Abm7 Db7 G%7 C7 Fm6. In cases in which a constituent is constituted by a strong prolongation (e.g., for the whole sequence of this example), we use the convention that the head is the right chord symbol. Since only the head of a complete constituent is allowed to refer to a chord outside the constituent, the concept of expectation-realization references is generalizable to complete constituents: we say that a complete constituent refers to a chord X if its head refers to X.

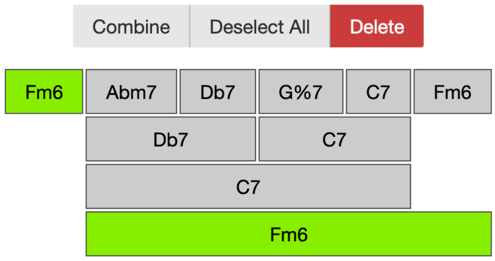


(a) Part of the harmonic syntax tree of *Birks's Works* from the treebank.

(b) Harmonic expectation-realization structure. This graph stands in 1-to-1 relation to the syntax tree shown in (a). Directed and undirected edges denote preparations and prolongations, respectively.

```
[.Fm6
    Fm6
    [.Fm6
        [.C7
            [.Db7
                Abm7
                Db7 ]
            [.C7
                G\%7
                C7 ] ]
        Fm6 ] ]
```

(c) String representation of the syntax tree in tikz-qtree format. This string is created using the tree annotation app shown in (d). The tree plot is shown in (a).



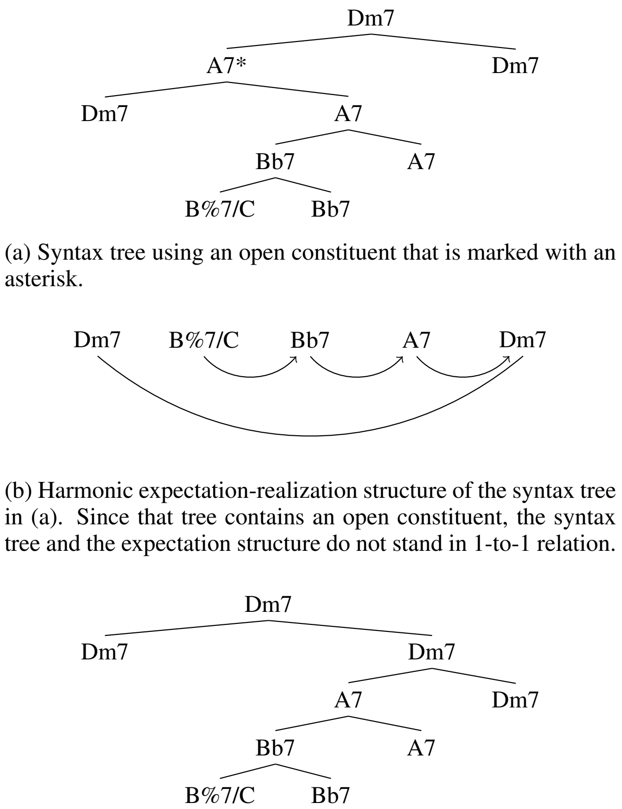
(d) Screenshot of tree annotation app. Each button represents a tree node. The user is selecting the green buttons to combine them to the full tree.

```
{"label": Fm6, "children": [
  {"label": "Fm6", "children": []},
  {"label": "Fm6", "children": [
    {"label": "C7", "children": [
      {"label": "Db7", "children": [
        {"label": "Abm7", "children": []},
        {"label": "Db7", "children": []}]},
      {"label: "C7", "children": [
        {"label: "G%7", "children": []},
        {"label: "C7", "children": []}]}]},
    {"label": "Fm6", "children": []}]}]}
```

(e) Tree string in JSON format, automatically converted from tikz-qtree format shown in (c).

Figure 1: Syntax tree of the final chords of the Jazz standard *Birk's works* in different representations.

[4] Note that the word head is used both for rules and constituents. This is not a problem since the head of a constituent is always the head of the top-most rule of its (sub-)tree analysis.



(a) Syntax tree using an open constituent that is marked with an asterisk.

(b) Harmonic expectation-realization structure of the syntax tree in (a). Since that tree contains an open constituent, the syntax tree and the expectation structure do not stand in 1-to-1 relation.

(c) Resolution of the open constituent in the syntax tree shown in (a). This tree stands in 1-to-1 relation to the expectation-realization structure in (b).

Figure 2: Hierarchical analysis of the initial chords of the Jazz standard *Why Don't You Do Right?* using open constituents (marked with asterisks).

In addition to complete constituents, one other constituent type is used in the JHT analyses. Consider for example the first four measures of the Jazz standard *Why Don't You Do Right?*,

| Dm7 B%7/C | Bb7 A7 | Dm7 B%7/C | Bb7 A7 |,

where B%7/C denotes a half-diminished seventh chord with root B and a C in the bass. The first two measures constitute a phrase following the *Lamento* schema (a stepwise descending movement of the bass from scale degree I to scale degree V [4]) that is repeated multiple times in the song. Since the transition from A7 to Dm7 does not sound like a resolution but more like a jump or an interruption (partly because of the repetition of the first two measures), we assume that A7 does not resolve into the following tonic Dm7, but into a tonic later in the song. Therefore, the phrase Dm7 B%7/C Bb7 A7 does constitute some kind of unit as shown in Figure 2a.

Since Dm7 and A7 both refer to a chord outside the phrase (see Figure 2b), the phrase does not have a head. It is therefore not a complete constituent. We call such constituents, in which multiple chords refer to a chord outside of the phrase, *open constituents*. The chords of an open constituent that refer to a chord outside of the constituent are called *chords with open references*. In the example of *Why Don't You Do Right?*, the chords Dm7 and A7 are the chords with open references of the open constituent Dm7 B%7/C Bb7 A7. Both chords Dm7 and A7 refer to the same tonic chord Dm7.

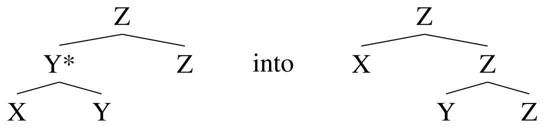
The JHT allows a single type of open constituent, called *restricted* open constituent, which consists of two adjacent constituents that refer to the same chord later in the piece. Since all constituents considered in the JHT are restricted in that way, we simple refer to them as open constituents. The restriction enables a further generalization of expectation-realization references to open constituents: We say that an open constituent refers to the chord to which all of its chords with open references refer. As shown in Figure 2a, the topmost node of an open constituent is labeled by the chord symbol of the right child of the node and additionally marked with an asterisk.

Other examples of open constituents are (i) I-VI-II-V-like phrases in *I Got Rhythm* and *I Can't Give You Anything But Love* and, in particular, (ii) tunes of form ABAC in which the B-part ends in a half cadence such as *All of Me*, *How High the Moon*, and *A Fine Romance*. *Summertime*, shown in Figure 3, is a prototypical example of a song with a ABAC form and a half cadence at the end of the B section. The interruption after the half cadence is supported by the movement from scale degree 3 to scale degree 2 in the melody and denoted using an open constituent.

2.3 Interpretation of Open Constituents as Prolongation-Preparation Structures

Syntax trees containing open constituents are interpretable as expectation-realization structures as shown in Figure 2. The interpretation procedure transforms a syntax tree that contains open constituents (e.g., Figure 2a) in to a tree that only represents prolongation and preparation operations (e.g., Figure 2c). This transformed tree then characterizes the expectation-realization structure (e.g., Figure 2b). Since open constituents are explicitly marked with asterisks, their interpretation is unambiguous.

To formalize the interpretation of open constituents, let Y^* be the chord symbol labeling an open constituent consisting of two constituents labeled with chord symbols X and Y. Let further be Z the chord symbol that is referenced by both X and Y. The reference is expressed by Z being the right sibling of the open constituent. The conversion then transforms



In the more general case of nested open constituents, the conversion is recursively applied from the root to the leaves of the tree (i.e., top-down).

The JHT contains trees for both representations, with open subtrees and in pure preparation-prolongation form. A python script was used to automatically transform the former into the letter. The script and additional utilities such as for tree traversal and drawing are provided with the treebank.

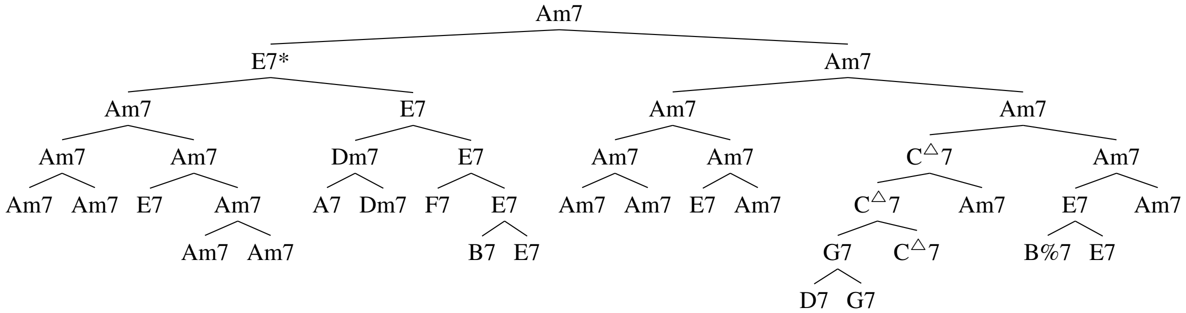

Am7
E7*
Am7
Am7
E7
Am7
Am7
Am7
Dm7
E7
Am7
Am7
C△7
Am7
Am7 Am7 E7 Am7 A7 Dm7 F7 E7 Am7 Am7 E7 Am7 C△7 Am7 E7 Am7
Am7 Am7 B7 E7 G7 C△7 B%7 E7
D7 G7

Figure 3: Complete syntax tree of the Jazz standard *Summertime* (turnaround omitted). The top levels of the tree reflect the ABAC form the song using an open constituent.

3. TREE ANNOTATION TOOL

The trees of the JHT are created using a graphical interface implemented as a simple web application, which was developed during the creation of the treebank. The source code of the application is written in ClojureScript (which compiles to JavaScript) and publicly available on GitHub. The application itself is hosted on GitHub pages and can be used independently of this dataset.[5] A screenshot of the application is shown in Figure 1d. The main part of the user interface displays a syntax tree that is represented by a hierarchical button layout. The user interface also contains an input-output section and buttons for creating, deleting, and deselecting tree nodes.

To create a syntax tree, the user inputs a sequence of space-separated strings such as chord symbols. To create an inner node of the tree, the nodes that become the child nodes of the new inner node are selected and combined by pressing a button or a key shortcut. Since the trees are mostly right-headed, the label of the rightmost child is used for the new node by default, but the label of a node can be changed arbitrarily. The output of the application is given as a string representation of the tree in tikz-qtree format as shown in Figure 1c and in JSON format as shown in Figure 1e.[6] Existing trees can be edited by loading them in any of these two formats. Since the application is designed to be agnostic to annotation conventions, it allows arbitrary labels and rule arities.

4. ANNOTATION PROCEDURE

All analyses in the dataset begin from chord sequences drawn from the iRealPro collection of Jazz standards. This collection was created by the user community of the iRealPro app[7] and transferred into kern format by Shanahan et al. [46].[8] We transformed the data into a JSON-like format and occasionally corrected individual chord symbols when we noticed serious differences between the iRealPro data and publicly available *Real Books* (i.e., collections of lead sheets.). Annotations of bass notes and optional chord tones such as ninths and elevenths were excluded from the chord symbols. Chord symbols with a duration of more than one measure were split into multiple chord symbols. 150 Jazz standards were selected for analysis (i) by filtering pieces that are within the scope of the theory of harmonic syntax described in Section 2 and (ii) by preferring shorter pieces. If applicable, turnarounds at the end of a lead sheet were deleted or a final tonic chord not contained in the lead sheet was added. All repetitions were unfolded and codas were appended at the positions indicated in the lead sheet. The selected Jazz standards were initially analyzed by the first author and a student assistant. The analyses were then reviewed by the second and the third author and discussed in the group. To ensure consistent analyses across all 150 Jazz standards, all final tree editing was performed by the first author.

Every hierarchical analysis denotes at least one author's mental representation of the harmonic structure of a Jazz standard. Each analysis is therefore also influenced by other musical features such as harmonic rhythm, phrasing, musical form, and melody. In ambiguous cases, the analyst chose the option that he seemed most important. These choices were necessary, because a single syntax tree can only encode one harmonic function for each chord. For example in the key C major, a C major triad can act as a tonic or as a preparation of a following F major chord. For five particularly ambiguous tunes, we provide alternative analyses in the treebank.

Since the iRealPro lead sheets were created and collected by the community of the application, the chord symbol usage is not fully consistent across the pieces. For instance, a Fm6 chord symbol can denote a tonic chord in F minor over a Dorian scale or a Bb9 chord with omitted root and fifth in the bass. Another example is that fourth-voicings are commonly denoted as suspension chords while actual suspensions of the scale degree V (e.g., suspension of C and E by B and D in a G major triad) are sometimes denoted as chords over the scale degree I (with or without explicitly mentioning the second inversion).

Furthermore, some chords do not have a proper harmonic function, but are better explained as voice-leading connections between two chords. The chords C C#o7 G/D at the beginning of the final 8 measures of *Bill Bailey* are an example of such a voice-leading connection (see Figure 4). Moreover, these final measures are an example of

[5] Link to tree annotation app: https://dcmlab.github.io/tree-annotation-code/

[6] https://www.ctan.org/pkg/tikz-qtree

[7] https://irealpro.com/

[8] The iRealPro dataset is available in kern format at http://doi.org/10.5281/zenodo.3546040.

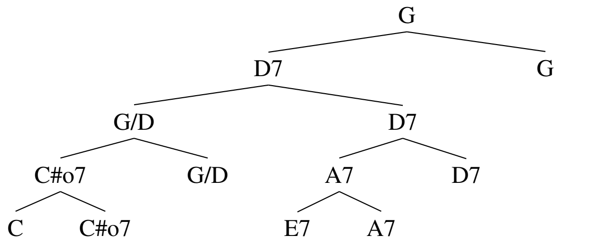


Figure 4: Syntax tree of the final 8 measures of *Bill Bailey* (turnaround omitted).

a common closing pattern. This pattern starts on the scale degree IV in its first measure, then transitions to a suspension of the scale degree V in measure 3, jumps away, and finally approaches the tonic through the cycle of fifths.

5. DATASET SUMMARY

The JHT is provided as a single file in JavaScript Object Notation (JSON) format. For each Jazz standard, this file contains the chord sequence with rhythmical information (measures and beats), metadata about title, composer(s), year of composition, time signature, and key (root & major/minor) as well as the tree analyses.[9]

In addition to the hierarchical analyses, some pieces contain a turnaround annotation represented as an integer. A value of zero means that the Jazz standard ends with a tonic chord. A positive value n means that the lead sheet of the piece ends with a turnaround of length n. For example, the chord sequence of *I love Paris* (in C major) has a turnaround length of $n = 2$, because it ends with the chords Dm7 G7 C6 D%7 G7. A negative turnaround annotation means that the tonic of the piece is not at the end of the piece, but at the beginning. A value of -1 indicates, for example, that the first chord of the chord sequence is the tonic of the piece, like in *Solar*. In rare cases, the tonic is not the first chord but the n-th chord which is represented by a turnaround annotation of $-n$.

The 150 chord sequences analysed in the treebank have an average length of 27.75 and consist of 11697 chords in total with 92 unique chord symbols. The syntax trees consist in total of 3899 binary rule applications with 512 unique rules and 268 open constituents. The average tree height is 7.57.

Further descriptive statistics of the JHT are visualized in Figure 5. The first plot shows that the subset of the analyzed pieces is chosen relatively independently from the year of composition. The second plot shows the bias for short pieces in this subset. The third plot shows that the length of turnarounds, if present, usually ranges between 1 and 3. The two last plots show separately for major and minor keys how often a context-free grammar rule is used in the hierarchical analyses. For these plots, all chord sequences were transposed to C major or to C minor, respectively. Prolongations of the tonic, preparations of the tonic by the fifth scale degree, and preparations of the fifth scale degree by the second are by far the most common rules.

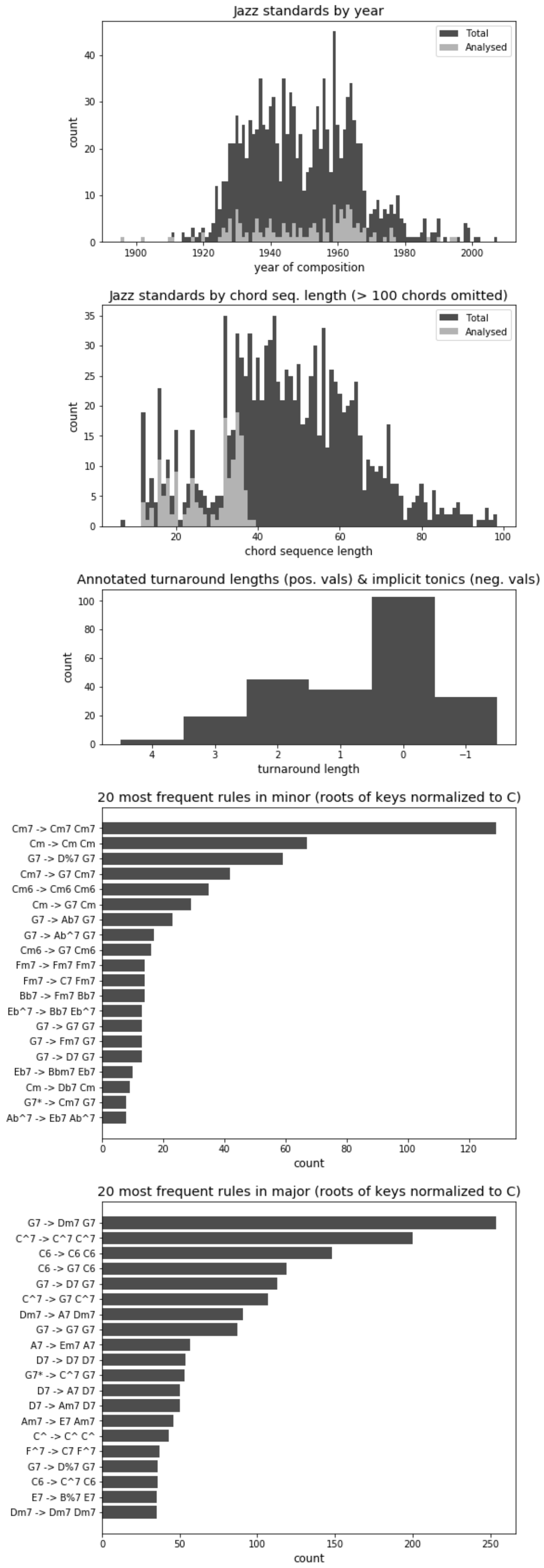


Figure 5: Plots of summary statistics of the tree analyses. See the main text for further explanation.

[9] The metadata was copied from the iRealPro dataset without detailed validity checking. It is provided for convenience.

6. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 760081 – PMSB. We gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds de Recherche du Québec, Société et Culture (FRQSC), and the Canada CIFAR AI Chairs program. We thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology. The authors additionally thank the anonymous referees for their valuable comments and the members of the Digital and Cognitive Musicology Lab (DCML) for fruitful discussions.

7. REFERENCES

[1] S. Abdallah, N. Gold, and A. Marsden. Analysing Symbolic Music with Probabilistic Grammars. In David Meredith, editor, *Computational Music Analysis*, pages 157–189. Springer International Publishing, Cham, 2016.

[2] J. A. Burgoyne, J. Wild, and I. Fujinaga. An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, Florida, 2011.

[3] A. Cadwallader and D. Gagné. *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press, Oxford, 2nd edition, 2007.

[4] W. E. Caplin. Topics and formal functions: The case of the lament. *The Oxford Handbook of Topic Theory*, page 415, 2014.

[5] T. Chen and L. Su. Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks. In *ISMIR*, pages 90–97, 2018.

[6] N. Chomsky. *Aspects of the Theory of Syntax*, volume 11. MIT press, 1965.

[7] W. B. De Haas, J. P. Magalhães, and F. Wiering. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 295–300, 2012.

[8] W. B. De Haas, J. P. Magalhães, F. Wiering, and R. C. Veltkamp. Automatic functional harmonic analysis. *Computer Music Journal*, 37(4):37–53, 2013. Publisher: MIT Press.

[9] J. Devaney, C. Arthur, N. Condit-Schultz, and K. Nisula. Theme And Variation Encodings with Roman Numerals (TAVERN). In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Malaga, Spain, 2015.

[10] C. Finkensiep, R. Widdess, and M. Rohrmeier. Modelling the Syntax of North Indian Melodies with a Generalized Graph Grammar. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 462–269. ISMIR, 2019.

[11] F. Foscarin, F. Jacquemard, P. Rigaux, and S. Masahiko. A parse-based framework for coupled rhythm quantization and score structuring. In *Mathematics and Computation in Music*, pages 248–260, Cham, 2019. Springer International Publishing.

[12] R. Gaizauskas. Investigations into the grammar underlying the Penn Treebank II. *Research Memorandum CS-95-25, Department of Computer Science, Univeristy of Sheffield*, pages 185–189, 1995.

[13] É. Gilbert and D. Conklin. A probabilistic context-free grammar for melodic reduction. In *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence*, pages 83–94, 2007.

[14] M. Gotham and M. T. Ireland. Taking Form: A Representation Standard, Conversion Code, and Example Corpus for Recording, Visualizing, and Studying Analyses of Musical Form. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[15] M. Granroth-Wilding and M. Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374, October 2014.

[16] R. Groves. Automatic Melodic Reduction Using a Supervised Probabilistic Context-free Grammar. *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016.

[17] M. Hamanaka, K. Hirata, and S. Tojo. Musical Structural Analysis Database based on GTTM. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014.

[18] D. Harasim, T. J. O'Donnell, and M. Rohrmeier. Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, 2019.

[19] D. Harasim, M. Rohrmeier, and T. J. O'Donnell. A Generalized Parsing Framework for Generative Models of Harmonic Syntax. *19th International Society for Music Information Retrieval Conference*, 2018.

[20] C. Harte, M. Sandler, S. Abdallah, and E. Gomez. Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations. In *Proceedings of the 6th International Society for Music Information Retrieval Conference*, London, UK, 2005.

[21] J. Hockenmaier and M. Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, 2007.

[22] J. Katz-Brown, S. Petrov, R. McDonald, F. Och, D. Talbot, H. Ichikawa, M. Seno, and H. Kazawa. Training a parser for machine translation reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 183–192, USA, 2011. Association for Computational Linguistics.

[23] P. B. Kirlin. *A Probabilistic Model of Hierarchical Music Analysis*. PhD thesis, University of Massachusetts Amherst, 2014.

[24] P. B. Kirlin and D. D. Jensen. Probabilistic Modeling of Hierarchical Music Analysis. In *ISMIR*, pages 393–398, 2011.

[25] S. Koelsch, M. Rohrmeier, R. Torrecuso, and S. Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, 2013.

[26] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA, 1983.

[27] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo, 2004.

[28] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. 1993.

[29] A. McLeod and M. Steedman. Meter Detection in Symbolic Music Using a Lexicalized PCFG. *Proceedings of the 14th Sound and Music Computing Conference*, 2017.

[30] A. McLeod and M. Steedman. Meter Detection and Alignment of MIDI Performance. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[31] G. Melis, C. Dyer, and P. Blunsom. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*, 2017.

[32] G. Micchi, M. Gotham, and M. Giraud. Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis. *Transactions of the International Society for Music Information Retrieval*, 3(1):42–54, May 2020.

[33] F. C. Moss, W. F. Souza, and M. Rohrmeier. Harmony and form in Brazilian Choro: A corpus-driven approach to musical style analysis. *Journal of New Music Research*, 2020.

[34] E. Nakamura, M. Hamanaka, K. Hirata, and K. Yoshii. Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[35] E. Nakamura, K. Itoyama, and K. Yoshii. Rhythm transcription of MIDI performances based on hierarchical Bayesian modelling of repetition and modification of musical note patterns. In *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016.

[36] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier. The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets. *Frontiers in Digital Humanities*, 5, July 2018.

[37] M. Neuwirth and M. Rohrmeier. Towards a syntax of the Classical cadence. In *What is a Cadence*, pages 287–338. Leuven University Press, 2015.

[38] J. Nivre. Dependency grammar and dependency parsing. *MSI report*, 5133(1959):1–32, 2005.

[39] D. Rizo and A. Marsden. A standard format proposal for hierarchical analyses and representations. In *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*, pages 25–32, 2016.

[40] M. Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[41] M. Rohrmeier. The syntax of jazz harmony: Diatonic tonality, phrase structure, and form. *Music Theory and Analysis*, 7(1):1–63, 2020. Publisher: Leuven University Press.

[42] M. Rohrmeier and I. Cross. Tacit tonality-Implicit learning of context-free harmonic structure. In *ESCOM 2009: 7th Triennial Conference of European Society for the Cognitive Sciences of Music*, 2009.

[43] M. Rohrmeier and M. Pearce. Musical syntax I: theoretical perspectives. In *Springer handbook of systematic musicology*, pages 473–486. Springer, 2018.

[44] A. Sarkar. Applying co-training methods to statistical parsing. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[45] H. Schenker. *Der Freie Satz. Neue musikalische Theories and Phantasien*. Universal edition edition, 1935.

[46] D. Shanahan, Y. Broze, and R. Rodgers. A Diachronic Analysis of Harmonic Schemata in Jazz. In *Proceedings of the 12th International Conference on Music*

Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music, pages 909–917, 2012.

[47] M. Steedman and J. Baldridge. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and explicit models of grammar*, pages 181–224, 2011.