A GENERALIZED PARSING FRAMEWORK FOR GENERATIVE MODELS OF HARMONIC SYNTAX

Daniel Harasim[1,2] **Martin Rohrmeier**[1,2] **Timothy J. O'Donnell**[3]

[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland

[2] Institut für Kunst- und Musikwissenschaft, TU Dresden, Germany

[3] Department of Linguistics, McGill University, Canada

daniel.harasim@epfl.ch

ABSTRACT

Modeling the structure of musical pieces constitutes a central research problem for music information retrieval, music generation, and musicology. At the present, models of harmonic syntax face challenges on the tasks of detecting local and higher-level modulations (most previous models assume a priori knowledge of key), computing connected parse trees for long sequences, and parsing sequences that do not end with tonic chords, but in turnarounds. This paper addresses those problems by proposing a new generative formalism Probabilistic Abstract Context-Free Grammars (PACFGs) to address these issues, and presents variants of standard parsing algorithms that efficiently enumerate all possible parses of long chord sequences and to estimate their probabilities. PACFGs specifically allow for structured non-terminal symbols in rich and highly flexible feature spaces. The inference procedure moreover takes advantage of these abstractions by sharing probability mass between grammar rules over joint features. The paper presents a model of the harmonic syntax of Jazz using this formalism together with stochastic variational inference to learn the probabilistic parameters of a grammar from a corpus of Jazz-standards. The PACFG model outperforms the standard context-free approach while reducing the number of free parameters and performing key finding on the fly.

1. INTRODUCTION

The modeling of non-local relations between musical objects such as notes and chords constitutes a central research problem for music information retrieval, music generation, and music analysis. Hierarchical models express these relations by assuming a latent hierarchical structure [19,22–24,30,31]. Consider for example the Jazz chord sequence Am^7 D^7 G^7 $C^{\triangle}$ where $C^{\triangle}$ denotes a major-seventh chord. Since the first three chords form a II V I sequence with reference to G^7 which is the dominant in C major, they form a *dominant phrase* [24]. The dominant phrase as a whole then refers to the tonic chord $C^{\triangle}$. All four chords together thus form a *tonic phrase*.

Figure 1 presents a syntactic analysis of the A-part of the Jazz-standard *Afternoon in Paris* following the approach from [22]. It illustrates the idea of how pieces can be decomposed into hierarchically-structured *constituents* which stand in part-whole relationship with one another. Subdominant, dominant, and tonic phrases are denoted by the scale degrees II, V, and I, respectively. Note that the subsequence Cm^7 F^7 $B\flat^{\triangle}$ is both a tonic progression in B♭ major and a dominant progression in E♭ major. It forms a dominant phrase in A♭ major together with $B\flat m^7$ and $E\flat^7$.

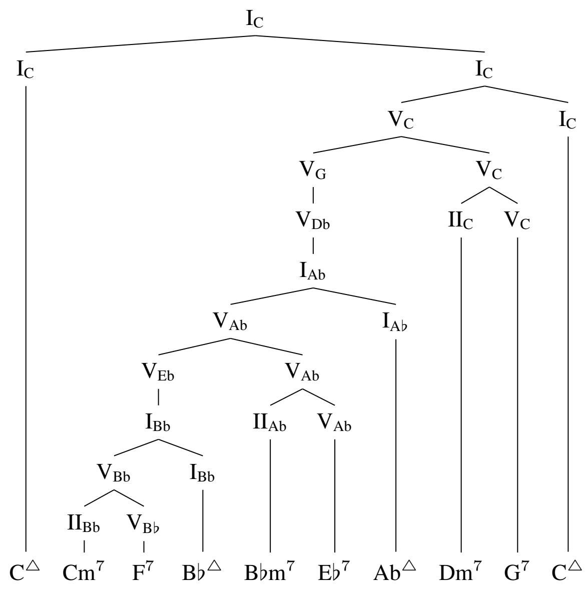

Figure 1. Hierarchical analysis of the A-part of the Jazz-standard *Afternoon in Paris*.

Models of harmonic syntax similar to Figure 1 have been successfully applied to melody harmonization [16], chord inference from audio [5, 6], and harmonic similarity [7]. There is also some empirical evidence for the psychological reality of hierarchical structures in music [15, 25]. While earlier theoretical and psychological work on hierarchical models has provided important insight about musical structure, computational implementation of these models to date has been limited to relatively small datasets. Earlier work includes applications to monophonic melodic data [21], a corpus of 39 blues chord progressions with a maximum of 24 chords per progression [12], or a dataset of 76 chord progressions (avg. length 40) from Jazz-standards that was restricted to subsequences of pieces that did not change key [4]. All these earlier approaches assume the knowledge of the key of the pieces a priori.

In computational linguistics, Context-Free Grammars (CFGs) are a standard way of modeling hierarchical constituent structure. They formalize constituent structures using *rewrite rules* denoted by long right arrows. The rule $X \longrightarrow Y\ Z$ for example states that the constituent X *consists of* the two constituents Y and Z. The existence of natural language treebanks makes it possible to read off the grammatical rewrite rules including their frequencies from syntactical analyses by experts. At present, there are music databases of simplified Schenkerian analyses [13], syntactic analyses of melodies based on the generative theory of tonal music [8], and annotated harmonic functions [4]. However, to the best of our knowledge there is currently no dataset of hierarchically analyzed chord sequences by human experts that could serve for the training or the evaluation of models of harmonic syntax. As a consequence, there exist no comparisons of models of harmonic syntax against expert analyses.

In the following, we introduce Abstract Context-Free Grammars (ACFGs), a generalization of the CFG framework designed to account for feature structures characteristic of musical categories. A first model of Jazz harmony is proposed in this framework that covers full pieces by incorporating modulations (i.e., changes in key). We train the model in a semi-supervised fashion on a dataset of Jazz-standards and evaluate it on a small set of hand-annotated hierarchical analyses. We further propose a solution for handling sequences that do not end with tonic chords, but in turnarounds. Simulations demonstrate that the ACFG model is able to outperform a PCFG model of the dataset. The implementation of the algorithms developed in this study are publicly available as a package of the Julia programming language [1]. [1]

2. OVERVIEW OF THE APPROACH

While the CFG framework has proven invaluable in computational linguistics, categories and part-whole relations between musical constituents have properties not possessed by linguistic structures. Musical categories such as scale degrees, for example, are equipped with an arithmetic structure that corresponds to musical transposition.

In the following, we refer to context-free rules of the form $X \longrightarrow Y\ X$ as a *preparation* of X by Y. The preparation of the scale degree $\mathrm{V}_{\mathrm{B}\flat}$ by $\mathrm{II}_{\mathrm{B}\flat}$ in *Afternoon in Paris* (see Figure 1) for example is a concrete realization of the general principle that any category x_k consisting of a scale degree x and a key k can be prepared by an ascending diatonic fifth $(x+4 \bmod 7)_k$. [24]. In addition to facts such as these, a framework for modeling musical structure has to account for the fact that the musical categories and rewrite rules are grouped into key-independent classes. For example, both $V_{\mathrm{B}\flat}$ and $V_{\mathrm{A}\flat}$ are fifth scale degrees. The probabilities of the application a rule to $V_{\mathrm{B}\flat}$ and $V_{\mathrm{A}\flat}$ should therefore be related.

This paper introduces Abstract Context-free Grammars (ACFGs), a modeling framework with a greater flexibility than CFGs. In particular, in ACFGs constituent categories are allowed to be of any data type and the rules are generalized partial functions. Unlike standard context-free rules, ACFG rules can therefore take advantage of the algebraic structure of categories. Probabilistic ACFGs extend probabilistic CFGs with the ability to express a wider range of probability distributions over rules.

3. ABSTRACT CONTEXT-FREE GRAMMARS

3.1 Definitions

Definition 1. A *(non-probabilistic) Abstract Context-free Grammar* (ACFG) $G = (T, C, C_0, \Gamma)$ consists of a set T of *terminal symbols*, a set C of *constituent categories*, a set of *start categories* $C_0 \subseteq C$, and a set of partial functions

$$\Gamma := \{\, r \mid r : C \nrightarrow (T \cup C)^* \,\},$$

called *rewrite rules* or *rewrite functions*. The arrow $\nrightarrow$ is used throughout the paper to denote partial functions. A sequence $\beta \in (T \cup C)^*$ can be *generated in one step* from a sequence $\alpha \in (T \cup C)^*$ by the application of a rewrite function $r \in \Gamma$, denoted by $\alpha \longrightarrow_r \beta$, if there exist $\alpha_1, \alpha_2 \in (T \cup C)^*$ and $A \in C$ such that $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 r(A) \alpha_2$. A sequence of rewrite rules $r_1 \dots r_n$ is called a *derivation* of a sequence of terminals $\alpha \in T^*$ if there exists a start category $\alpha_1 \in C_0$, and $\alpha_2, \dots, \alpha_n \in (C \cup T)^*$ such that

$$\alpha_1 \longrightarrow_{r_1} \alpha_2 \longrightarrow_{r_2} \cdots \longrightarrow_{r_n} \alpha,$$

where r_i is always applied to the leftmost category of α_i for $i \in \{1, \dots, n-1\}$. The set of derivations of α is denoted by $D(\alpha)$. The language of the grammar G is the set of terminal sequences that have a derivation in G.

Note that if C is finite, the languages that can be described by ACFGs are exactly the languages that can be described by standard context-free grammars (CFGs). For each ACFG with finite C, a CFG with rule set R can be constructed by dividing each rewrite function with domain cardinality k into k standard context-free rewrite rules,

$$R := \bigcup_{r \in \Gamma} \{\, (A, \alpha) \in C \times (T \cup C)^* \mid r(A) = \alpha \,\}.$$

Definition 2. A *Probabilistic Abstract Context-free Grammar* (PACFG) is an ACFG where each category $A \in C$ is associated with a random variable X_A over rewrite functions r such that $\mathbb{P}(X_A = r)$ is positive if and only if $r(A)$

[1] https://github.com/dharasim/GeneralizedChartParsing.jl

is defined, that is A is in the domain of r, $A \in \text{dom}(A)$. The probability $p(d)$ of a derivation $d = r_1 \dots r_n$ of a sequence of terminal symbols $\alpha \in T^*$ is defined as the product $\prod_{i=1}^{n} \mathbb{P}(X_{A_i} = r_i)$ where in each step r_i is applied to a category $A_i \in C$. The probability of α is then defined as $p(\alpha) = \sum_{d \in D(\alpha)} p(d)$.

Note that PACFG categories can share the same probability distribution over rewrite functions without rewriting to exactly the same right-hand sites. This important property allows us to model the structural relations between musical keys. We use this property in Section 4 to build a model that abstract chords sequences from their concrete scale by defining the probability that a rewrite function is applied to a scale degree independently of its key. The sharing of probability mass between rules additionally reduces the number of free parameters of a PACFG model.

To illustrate the different learning capabilities of PCFG and PACFG models, consider a toy PCFG with nonterminal symbols $C = \{ S, A, B \}$, start symbol S, terminal symbols $T = \{ a, b \}$, and rules $S \longrightarrow A \mid B$, $A \longrightarrow A\ A \mid a$, and $B \longrightarrow B\ B \mid b$. The grammar thus generates sequences that solely consist either of as or bs. In a classical PCFG setting, no probability mass is shared between rules, but each rule has its separate probability. However, in the process of inferring the probabilities of the rules from data, it might be desirable to generalize the rules $A \longrightarrow A\ A$ and $B \longrightarrow B\ B$ to a meta rule $x \longrightarrow x\ x$ where $x \in \{ A, B \}$ and to put probability mass on this abstract entity. In that way, the grammar can learn something about $A \longrightarrow A\ A$ when it observes $B \longrightarrow B\ B$ and vice versa. The PACFG version of the PCFG presented above addresses the problem by replacing the classical context-free rules by the partial functions r_1, r_2, r_3, r_4, and r_5 with $r_1(S) = A$, $r_2(S) = B$, $r_3(x) = x\ x$ for $x \in \{ A, B \}$, $r_4(A) = a$, and $r_5(B) = b$. Analogously, a PACFG of Jazz chord sequences can generalize classical rewrite rules so that their probabilities do not depend on the keys of their left-hand sides to model transpositional invariance.

3.2 Parsing

Parsing a sequence of terminal symbols with respect to a formal grammar is the task of computing the distribution of parse trees conditioned on this sequence. Many parsers are based on versions of the CYK algorithm that assumes grammars to be given in Chomsky normal form. Since grammar transformations into Chomsky normal form considerably blow up the grammar, the here presented parser transforms grammars on the fly during parsing, similar to the transformation presented in [18]. Each rule of the form $A \longrightarrow B_1 \dots B_k$ is transformed into a set of *states* $s_i = B_1 \dots B_i$ for $1 \leq i \leq k$, a transition function

$$\text{tran} : S \times (T \cup C) \to S, \quad \text{tran}(s_i, B_{i+1}) = s_{i+1}$$

and a completion function $\text{comp} : S \to 2^C$ such that $\{ A \} \subseteq \text{comp}(s_k)$, where S denotes the set of all states. Note that the states and the transition function form a search trie where the completion function checks if there

items: edges $[s, i, j]$ for $s \in S$
constituents $[A, i, j]$ for $A \in C$
for and $i, j \in \{1, \dots, |\alpha| + 1\}$

goal items: $[A, 1, |\alpha| + 1]$ for $A \in S$

axioms: $$\frac{}{[\alpha_i, i, i+1]} \quad \text{for } i \in \{1, \dots, |\alpha|\}$$

introduce edge: $$\frac{[A, i, j]}{[s, i, j]} \quad s = \text{tran}(s_0, A)$$

complete edge: $$\frac{[s, i, j]}{[A, i, j]} \quad A \in \text{comp}(s)$$

fundamental rule: $$\frac{[s, i, j] \quad [A, j, k]}{[s', i, k]} \quad \text{tran}(s, A) = s'$$

Figure 2. Description of the parsing algorithm in the parsing as deduction framework. Existing Constituents can start the parser to read a sequence of terminal symbols and categories by the *introduce edge* rule. The *fundamental rule* is then recursively applied to extend these sequences. The *complete edge* rule eventually merges sequences to single constituents if they are the right-hand side of a grammar rule.

is a rewrite rule that has a sequence of terminal symbols and categories as its right-hand side. This trie data structure leads to a compact representation of the forest of all trees for a given input sequence. More generally, the parser can handle any transition and completion functions derived from finite-state automata, see [14].

In the following, a generic bottom-up parsing algorithm for abstract grammars is presented in the parsing as deduction framework using the above defined transition and completion functions [3, 29]. The parsing as deduction framework is a meta-formalism to state and compare different parsing algorithms. It views the parses of a sequence as logical deductions of goals from axioms by using constituents as atomic logical formulas. The formula $[\mathrm{I}_{\mathrm{B}\flat}, 2, 5]$ for example states the existence of a constituent with category $\mathrm{I}_{\mathrm{B}\flat}$ that spans over the second, third, and fourth terminal symbol. This formula is true in the analysis presented in Figure 1 because that analysis contains a constituent with label $\mathrm{I}_{\mathrm{B}\flat}$ over the span from the second to the forth leaf chord. The goals are constituents that span the full sequence and come from the set of start categories. The axioms are formulas of the form $[t_i, i, i+1]$ for each terminal in the input sequence $t_1 \dots t_n$. The parsing strategy such as bottom-up parsing or Earley parsing is encoded in the deduction rules. These rules are denoted by a set of atomic formulas over a horizontal line, an atomic formula under this line, and an optional side condition (see Figure

2). The formula under the line can be deduced from the formulas above if the side condition holds.

The proposed algorithm makes use of two different kinds of atomic formulas: edges (not yet completed constituents) and constituents. A state $s \in S$ together with a start index i and an end index j is called an *edge* and denoted by $[s, i, j]$. Analogously, a category $A \in C$ together with start and end indices i and j is called a *constituent* and denoted by $[A, i, j]$. Figure 2 shows the axioms, goal items, and the deduction rules of our algorithm.

3.3 Inference of Rule Probabilities

In this section, we give an overview of an inference algorithm for the rule probabilities $\mathbb{P}(X_A = r)$. Let $\Gamma_A = \{ r \in \Gamma \mid A \in \text{dom}(r) \}$ be the set of rewrite functions whose domain contains the constituent category A. We place a Dirichlet distribution on the probability vector describing the distribution over Γ_A, $\vec{\theta}_{\Gamma_A} \sim \text{Dirichlet}(\vec{\alpha}_{\Gamma_A})$ for pseudocount vector $\vec{\alpha}_{\Gamma_A}$. The inference problem is to compute the posterior distribution over this set of probability vectors, given the data D and pseudocounts $\{\vec{\alpha}_{\Gamma_A}\}$,

$$p(\{\vec{\theta}_{\Gamma_A}\} \mid D, \{\vec{\alpha}_{\Gamma_A}\}) \propto p(D \mid \{\vec{\theta}_{\Gamma_A}\})p(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\alpha}_{\Gamma_A}\}),$$

where $\{\vec{\theta_{\Gamma_A}}\}$ is an abbreviation for $\{\vec{\theta_{\Gamma_A}}\}_{A \in C}$, etc. *Variational Bayesian inference* (VB) is used to approximate this posterior distribution [2, 11, 32]. We introduce an approximating *variational distribution* $q(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\nu}_{\Gamma_A}\})$ with *variational parameters* $\{\vec{\nu}_{\Gamma_A}\}$ over our target hidden variables (rule weights) and minimize the Kullback-Leibler divergence between this approximation and the true posterior,

$$D_{\text{KL}}(q(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\nu}_{\Gamma_A}\}) \mid\mid p(\{\vec{\theta}_{\Gamma_A}\} \mid D, \{\vec{\alpha}_{\Gamma_A}\})),$$

by adjusting the variational parameters $\{\vec{\nu}_{\Gamma_A}\}$.

Following [17], we approximate the distribution over each probability vector with a Dirichlet distribution $\vec{\theta}_{\Gamma_A} \mid \vec{\nu}_{\Gamma_A} \sim \text{Dirichlet}(\vec{\nu}_{\Gamma_A})$, and make use of the *mean-field approximation*

$$q(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\nu}_{\Gamma_A}\}) = \prod_{A \in C} p(\vec{\theta}_{\Gamma_A} \mid \vec{\nu}_{\Gamma_A}).$$

We minimize the Kullback-Leibler divergence with a coordinate descent algorithm similar to the expectation-maximization algorithm. First, we compute the expectation of the counts of rule usages in the data under our current setting of the variational parameters, $\mathbb{E}_q[\#(r, D)]$ where $\#(r, D)$ is the number of times that rule r was used to generate the data D, and then we update our variational parameters based on these expectations. Since all of our distributions are in the exponential family, it can be shown that the optimal update is given by the equation $\hat{\vec{\nu}}_{\Gamma_A} = \vec{\alpha}_{\Gamma_A} + \mathbb{E}_q[\#(r, D)]$ [2]. In other words, we set the pseudocounts of our variational distributions equal to the expected number of rule usages plus the pseudocount for each rule in the prior distribution.

Under the standard coordinate-ascent algorithm given in [17], expected counts must be computed for the whole corpus before updating using the equation above. Hoffman et al. [9] propose a stochastic variant of the standard variational (inspired by *stochastic gradient descent*) where updates are computed with respect to randomly sampled *minibatches* of the data. We make use of this *stochastic variational Bayes* algorithm in the results reported below.

4. A GENERATIVE MODEL OF JAZZ HARMONY

This section presents a PACFG $G = (T, C, C_0, \Gamma)$ that models the syntax of Jazz harmony following the proposal in [24]. That work addressed the problem of finding a restrictive grammar that describes the full variety of syntactic relations in the musical idiom of Jazz-standards. The set of terminal symbols T is a set of pairs describing chords each of which consists of the root of the chord and a string describing the chord form—one of: a major triad, a major-seventh chord, a major sixth chord, a dominant-seventh chord, a minor triad, a minor-seventh chord, a half-diminished-seventh chord, a diminished seventh-chord, an augmented triad, or a suspended chord.

In the following, $\mathbb{Z}_n$ denotes the ring of integers modulo $n \in \mathbb{N}$. The categories are modeled as pairs of scale degrees and keys, $C = \mathbb{Z}_7 \times K$, where a key consists of a pitch class representing its root and a string describing its mode, $K = \mathbb{Z}_{12} \times \{ \text{major}, \text{min} \}$. Scale degrees are denoted by roman numerals from I to VII. All categories with scale degree I are start symbols, $C_0 = \{ \text{I} \} \times K$. Let $k \in K$ denote an arbitrary key. The set of rewrite functions Γ consists of *prolongation*,

$$\text{PROLONG}(\langle x, k \rangle) = \langle x, k \rangle \; \langle x, k \rangle$$

for $x \in \mathbb{Z}_7$, *diatonic preparation*,

$$\text{DIAT-PREP}(\langle x, k \rangle) = \langle x + 4 \bmod 7, k \rangle \; \langle x, k \rangle$$

for $x \in \mathbb{Z}_7 \setminus \{ \text{IV} \}$, *dominant preparation*,

$$\text{DOM-PREP}(\langle x, k \rangle) = \langle \text{V}, \mu(x, k) \rangle \; \langle x, k \rangle$$

for $x \in \mathbb{Z}_7 \setminus \{ \text{I} \}$ where $\mu(x, k)$ denotes the modulation from k into the key of scale degree x (e.g. $\mu(\text{II}, (0, \text{maj})) = (2, \text{min})$, the key of the second scale degree of C major is D minor), *plagal preparation*,

$$\text{PLAGAL-PREP}(\langle \text{I}, k \rangle) = \langle \text{IV}, k \rangle \; \langle \text{I}, k \rangle,$$

modulation,

$$\text{MODULATION}(\langle x, k \rangle) = \langle \text{I}, \mu(x, k) \rangle,$$

mode change,

$$\text{MODE-CHANGE}(\langle \text{I}, (r, m) \rangle) = \begin{cases} \langle \text{I}, (r, \text{min}) \rangle, & \text{if } m = \text{maj} \\ \langle \text{I}, (r, \text{maj}) \rangle, & \text{if } m = \text{min}, \end{cases}$$

for $r \in \mathbb{Z}_{12}, m \in \{ \text{maj}, \text{min} \}$, *diatonic substitution*,

$$\text{DIAT-SUBST}(\langle x, (r, m) \rangle) = \begin{cases} \langle \text{VI}, (r, m) \rangle, & \text{if } x = \text{I}, m = \text{maj} \\ \langle \text{III}, (r, m) \rangle, & \text{if } x = \text{I}, m = \text{min} \\ \langle \text{IV}, (r, m) \rangle, & \text{if } x = \text{II} \\ \langle \text{VII}, (r, m) \rangle, & \text{if } x = \text{V} \end{cases}$$

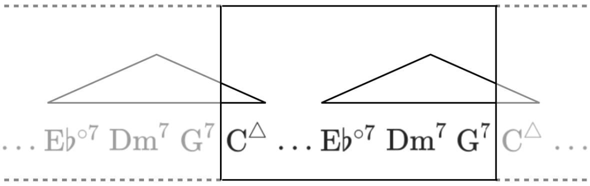

Figure 3. Parsing the turnaround of *All of me*

for $x \in \{\, \mathrm{I}, \mathrm{II}, \mathrm{V} \,\}$, $r \in \mathbb{Z}_{12}$, $m \in \{\, \mathrm{maj}, \mathrm{min} \,\}$, and *dominant substitution*,

$$\textsc{dom-subst}_i(\langle \mathrm{V}, (r, m) \rangle) = \langle \mathrm{V}, (r + i \bmod 12, m) \rangle$$

for $r \in \mathbb{Z}_{12}$, $m \in \{\, \mathrm{maj}, \mathrm{min} \,\}$, and $i \in \{\, 3, 6, 9 \,\}$. Additionally, Γ contains appropriate termination rules $C \twoheadrightarrow T$ according to standard Jazz harmony theory (e.g. seventh-chord-termination$(\langle 4, (0, \mathrm{maj}) \rangle) = \mathrm{G}^7$, see [20] for further explanation). The distribution of $X_{\langle x,k \rangle}$ over rules rewriting the category $\langle x, k \rangle$ is defined as a categorical distribution such that $\mathbb{P}(X_{\langle x,k \rangle} = r) = \mathbb{P}(X_{\langle x,k' \rangle} = r)$ for all scale degrees x, rules r, and keys k, k' that have the same mode. That is, the probability of r rewriting $\langle x, k \rangle$ does not depend on the root of k which enables the model to learn the parameters of its probability distributions key-independently.

These grammar rules can be grouped into three classes: the prolongation rule, preparation rules, and substitution rules. Preparation rules create categories that for the listener generate the expectation to hear the prepared chord. Substitution rules substitute chords for other chords that fulfill an equivalent function inside the sequence such as tritone substitutions of dominants in Jazz.

5. THE TURNAROUND PROBLEM

A lead-sheet of a Jazz-standard consists of a melody together with a chord sequence describing the fundamental harmonic structure of the piece. The chord sequence is repeated multiple times in a performance. While some lead-sheets end with tonic chords, others include harmonic upbeats to the first chord of the piece at the end of the sheet, called *turnarounds*. The final chord of a performances is nevertheless usually a tonic chord. The lead-sheet of the Jazz-standard *All of me* starts for example with a $\mathrm{C}^{\triangle}$ chord and ends with the turnaround $\mathrm{E}\flat^{\circ 7}$ Dm^7 G^7.

The grammar of Jazz harmony proposed above assumes that pieces end with a tonic chord. Therefore, a simple implementation of this grammar would not able to parse lead-sheets that end in turnarounds. We solve this problem by *cyclic parsing*, meaning that we assume that constituents can have spans from the end of a piece back to the beginning, see Figure 3.

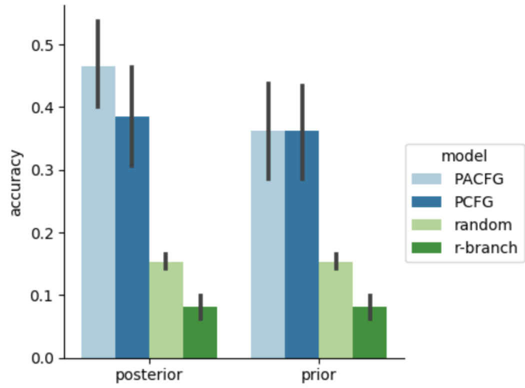

Figure 4. Tree accuracy plot

6. EXPERIMENTS

6.1 Dataset

The model is evaluated using the *iRealPro* dataset of Jazz-standards. [2] This dataset consists of 1173 chord sequences electronically-encoded by the Jazz musician community including metadata such as the titles, composers, and keys. The sequences were collected and converted into the Humdrum format [10] by Daniel Shanahan and Yuri Broze [28], and are available online. [3] For other research that uses this dataset see [26, 27]. The chord forms in the *iRealPro* dataset include information about nineths and elevenths that are not considered in this study.

The subset of 394 Jazz-standards that consist of at most 40 chords was considered to train the models. 34.52% (136) of these pieces were parsable using the standard approach and 90.61% (357) pieces were parsable using the cyclic parsing approach described above. Less then 55% of the considered Jazz-standards therefore end in turnarounds.

6.2 Tree Accuracy Evaluation

We compare four models: (i) the proposed PACFG model that uses a representation of rules independent of key, (ii) its PCFG counterpart the rules of which are not independent of key, (iii) a baseline of randomly generated trees, and (iv) a *right-branching baseline* in which all constituents split into a constituent on the left and a terminal symbol on the right.

The models are trained on the 357 cyclic parsable sequences using minibatches of 8 sequences. They are evaluated on 13 pieces hand-annotated by the authors. We report the predicted tree accuracy. That is the precision of correctly predicted spans of internal tree nodes. A span of a tree node is defined as the start index of its leftmost leaf together with the end index of its rightmost leaf.

Figure 4 shows the means of the tree accuracies including 95% confidence intervals as error bars. The right-branching baseline performs at an accuracy level under 10%. The random baseline performs slightly better at an

[2] https://irealpro.com
[3] https://musiccog.ohio-state.edu/home/index.php/iRb_Jazz_Corpus

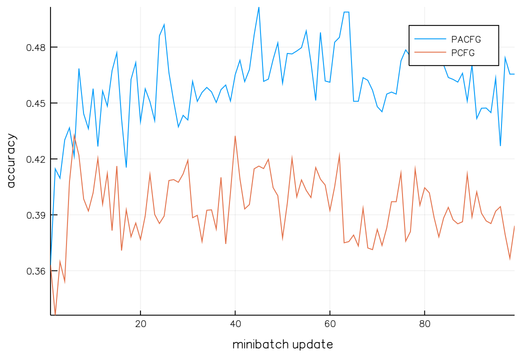

Figure 5. Predicted tree accuracy for each minibatch update. Note that the y-axis displays only values between 33% and 50%.

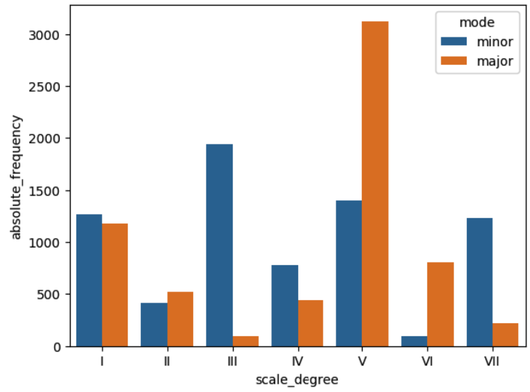

Figure 6. Expected usage of scale degrees to parse the full training dataset

accuracy level of 15.35% Under a uniform prior, both the PACFG and the PCFG model perform at an accuracy level of 36.30% a priori of the data. As opposed to the trained PCFG model that only improves its performance by about 3% (in comparison to the uniform prior) reaching an accuracy of 39.43%, the trained PACFG model improves by about 10% (in comparison to the uniform prior) reaching an accuracy of 45.95%. The PACFG model was thus able to learn more from the data than the PCFG model. Note that since the PCFG model does not abstract the grammar rules from the concrete key wherein they are applied, the number of free parameters of the PCFG model is approximately 12 times higher than the number of free parameters of the APCFG model.

Despite the fact that the PACFG model learns key-independently, it is still much simpler than models that produce state-of-the-art parsing results in computational linguistics. In particular, state-of-the-art models in computational linguistics typically make use of conditioning information beyond the parent constituent categories used in the PACFG model—such as larger tree fragments, conditioning on heads and/or adjacent elements in the string, state-splitting, and other richer contextual information. We anticipate that the inclusion of similar structures into musical parsing models will lead to similar improvements in performance.

Figure 5 shows the mean predicted tree accuracies of the PACFG and the PCFG models for each minibatch update. Note that this figure is produced using a stochastic algorithm and is therefore inherently noisy. We see that the stochasticity of the inference algorithm leads to random jumps of the accuracy up to 0.5%. The models appear to do most of their learning in the first 10 minibatches.

6.3 Performance Diagnosis using Scale Degree Frequencies

Figure 6 shows the expected frequency of scale-degree use in the whole corpus. The scale degrees VI in major and III in minor are more frequently used by the model than expected. Because these scale degrees are substitutions for the first scale degrees and because they enable modulations into the relative key (e.g. from C major to A minor and vice versa), the model may be using them to alternate between relative keys. The prominence of the VII in minor keys is probably related to the fact that it has a dominant-seventh chord form. The model may be interpreting a I in major as a III in the relative minor key that is then prepared by the VII in minor. For example, the simple chord transition G^7 $\mathrm{C}^{\triangle}$ would in this case be derived by

$$\mathrm{I}_a \longrightarrow \mathrm{III}_a \longrightarrow \mathrm{VII}_a\ \mathrm{III}_a \longrightarrow \mathrm{G}^7\ \mathrm{III}_a \longrightarrow \mathrm{G}^7\ \mathrm{C}^{\triangle}.$$

7. CONCLUSION AND FUTURE RESEARCH

The research presented here introduced a new general grammar and parsing framework tailored to the needs of music and showed how to perform inference for such a model.

Experiments show that in contrast to standard context-free models, the proposed model is able to learn characteristic structures of the observed data. To the best of our knowledge, this is the first computational approach that automatically performs hierarchical analyses of chord sequences and evaluates them on analyses by human experts.

This paper lays the groundwork for more advanced models of harmonic syntax. Our future research will in particular focus on expanding the dataset of hand-annotated expert analyses to provide significance tests of the performance comparison of different models, for example. Further studies can use the tools developed here to build models of unsupervised grammar induction, joint models of multiple musical levels of musical structure like harmony and rhythm, and models of musical structure that have more complex dependencies than those representable in simple tree structures.

8. ACKNOWLEDGEMENTS

Financial support for the research presented in this article has in part been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the *Zukunftskonzept* at TU Dresden funded by the *Exzelleninitiative* of the *Deutsche Forschungsgemeinschaft*.

9. REFERENCES

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM review*, 59(1):65–98, 2017.

[2] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational Inference: A Review for Statisticians. *arXiv*, (arXiv:1601.00670), 2017.

[3] Joshua T Goodman. *Parsing inside-out*. PhD thesis, 1998.

[4] Mark Granroth-Wilding and Mark Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374, 10 2014.

[5] W Bas De Haas. *Music information retrieval based on tonal harmony*. PhD thesis, 2012.

[6] W Bas De Haas, Jos Pedro Magalhães, and Frans Wiering. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. *International Society for Music Information Retrieval Conference (ISMIR)*, (Ismir):295–300, 2012.

[7] W Bas De Haas, Martin Rohrmeier, and Frans Wiering. Modeling Harmonic Similarity using a Generative Grammar of Tonal Harmony. *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, 2009.

[8] Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Musical Structural Analysis Database Based on Gttm. In *Proceedings of the 15th Conference of the International Society for Music Information Retrieval*, pages 325–330, 2014.

[9] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[10] David Brian Huron. *The Humdrum Toolkit: Reference Manual*. Center for Computer Assisted Research in the Humanities, 1994.

[11] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakola, and Lawrence K Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37:183–233, 1999.

[12] Jonah Katz. Harmonic Syntax of the Twelve-Bar Blues Form. *Music Perception: An Interdisciplinary Journal*, 35(2):165–192, 2017.

[13] Phillip B Kirlin and David D Jensen. Using Supervised Learning to Uncover Deep Musical Structure. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1770–1776, 2015.

[14] Dan Klein and Christopher D. Manning. Parsing and Hypergraphs. *Proceedings of the 7th International Workshop on Parsing Technologies (IWPT-2001)*, (c):351372, 2001.

[15] Stefan Koelsch, Martin Rohrmeier, Renzo Torrecuso, and Sebastian Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, 2013.

[16] Hendrik Vincent Koops, Jos Pedro Magalhães, and W. Bas de Haas. A functional approach to automatic melody harmonisation. In *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design - FARM '13*, page 47. ACM Press, 2013.

[17] Kenichi Kurihara and Taisuke Sato. An Application of the Variational Bayesian Approach to Probabilistic Context-Free Grammars. 2004.

[18] Martin Lange and Hans Leiß. To CNF or not to CNF - An Efficient Yet Presentable Version of the CYK Algorithm. *Informatica Didactica*, 8:1–21, 2009.

[19] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA, 1983.

[20] Mark Levine. *The jazz theory book*. Sher Music, 1995.

[21] Eita Nakamura, Masatoshi Hamanaka, Keiji Hirata, and Kazuyoshi 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)*, pages 276–280, 2016.

[22] Markus Neuwirth and Martin Rohrmeier. Towars a syntax of the Classical cadence. In *What is a Cadence*, pages 287–338. 2015.

[23] Martin Rohrmeier. A generative grammar approach to diatonic harmonic structure. *Proceedings SMC'07, 4th Sound andMusic Computing Conference*, (July):11–13, 2007.

[24] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 3 2011.

[25] Martin Rohrmeier and Ian Cross. Tacit tonality : Implicit learning of context-free harmonic structure. In *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyväskylä, Finland*, number Escom, pages 443–452, 2009.

[26] Keith Salley and Daniel T. Shanahan. Phrase Rhythm in Standard Jazz Repertoire: A Taxonomy and Corpus Study. *Journal of Jazz Studies*, 11(1):1, 2016.

[27] Daniel Shanahan and Yuri Broze. Diachronic Changes in Jazz Harmony: A Cognitive Perspective. *Music Perception: An Interdisciplinary Journal*, 31(1):32–45, 2013.

[28] Daniel Shanahan, Yuri Broze, and Richard 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.

[29] Stuart M Shieber, Yves Schabes, and Fernando C.N. Pereira. Principles and Implementation of Deductive Parsing. *Journal of Logic Programming*, 1993.

[30] Mark J. Steedman. A Generative Grammar for Jazz Chord Sequences. *Music Perception: An Interdisciplinary Journal*, 2(1):52–77, 1984.

[31] Mark J Steedman. The blues and the abstract truth: Music and mental models. *Mental models in cognitive science: essays in honour of Phil Johnson-Laird*, pages 305–318, 1996.

[32] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.