



ChordRipple: Recommending Chords to Help Novice Composers Go Beyond the Ordinary

The Harvard community has made this
article openly available. [Please share](#) how
this access benefits you. Your story matters

Citation	Huang, Cheng-Zhi Anna, David Duvenaud, and Krzysztof Z. Gajos. 2016. ChordRipple: Recommending Chords to Help Novice Composers Go Beyond the Ordinary. In Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI '16), Sonoma, CA, March 7-10, 2016: 241-250.
Citable link	http://nrs.harvard.edu/urn-3:HUL.InstRepos:29405812
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP

ChordRipple: Recommending Chords to Help Novice Composers Go Beyond the Ordinary

Cheng-Zhi Anna Huang
Harvard University
czhuang@fas.harvard.edu

David Duvenaud
Harvard University
dduvenaud@seas.harvard.edu

Krzysztof Z. Gajos
Harvard University
kgajos@seas.harvard.edu

ABSTRACT

Novice composers often find it difficult to go beyond common chord progressions. To make it easier for composers to experiment with radical chord choices, we built a creativity support tool, CHORDRIPPLE, which makes chord recommendations that aim to be both diverse and appropriate to the current context. Composers can use it to help select the next chord, or to replace sequences of chords in an internally consistent manner.

To make such recommendations, we adapt a neural network model from natural language processing known as WORD2VEC to the music domain. This model learns chord embeddings from a corpus of chord sequences, placing chords nearby when they are used in similar contexts. The learned embeddings support creative substitutions between chords, and also exhibit topological properties that correspond to musical structure. For example, the major and minor chords are both arranged in the latent space in shapes corresponding to the circle-of-fifths.

Our structured observations with 14 music students show that the tool helped them explore a wider palette of chords, and to make “big jumps in just a few chords”. It gave them “new ideas of ways to move forward in the piece”, not just on a chord-to-chord level but also between phrases. Our controlled studies with 9 more music students show that more adventurous chords are adopted when composing with CHORDRIPPLE.

Author Keywords

Music; Songwriting; Chords; Harmony; Embeddings; Creativity Support Tools, Recommender Systems, Neural Language Models

ACM Classification Keywords

H.5.5 [Sound and Music Computing]: Modeling, Systems

INTRODUCTION

Novice composers often find themselves consciously or unconsciously falling back to familiar sets of chords, because these chords are what they are “used to hearing” and where their fingers are used to falling on their instrument. In the corpus of 200 rock songs curated from the *Rolling Stone Top 500 Hits* [14], we see a long-tail phenomenon where a few chord

progressions are used in many songs while most other chord progressions are rarely used, akin to the Zipf distribution of words in natural languages. In fact, a recent article showed that the evolution of our harmonic language, among other musical facets in contemporary western popular music, has stagnated for more than fifty years [11].

However, the challenge is often not only to find something that is novel, but to use these chords in a way that serves a personal musical impetus. In practice, novice composers often find it difficult to use more elaborate chords because they are less certain about their effects and how to integrate them into their own music in a way that feels both original and connected.

To address this challenge, we are developing CHORDRIPPLE (shown in Figure 1 and short-handed as CR)—a system that helps composers be more adventurous by recommending chords that are similar to the chords that they are using thus preserving the composer’s original musical intent, but that are a little more unusual. By proposing such chords as possible alternatives to the composer’s current chord choice, the system helps the composer see less familiar chords in the light of their own musical discourse, allowing them to easily experiment with the alternatives by substituting them into the current musical context.

We introduce a novel intervention called RIPPLE, which are chord recommendations that not only suggests alternative chords to the current chord but also recommends a reworking of its content. The name RIPPLE is an conceptual analogy to rippling effects, which in this case is the changing of one chord causing the change of its surrounding chords.

Chord recommendations are enabled by a machine learning model that captures chord similarity based on how likely different chords are to be used in similar musical contexts. We trained this model on a corpus of 200 rock songs curated from the *Rolling Stone Top 500 Hits* [14]. This model allows the system to recommend chords that are novel for a particular musical context but still similar to the chords used by the composer.

To evaluate how CR impacts composers’ creative process, we recruited 14 undergraduate music students, and asked them to use CR to further develop chord progressions for a piece they were working on. We observed both positive and negative impacts of the tool on the creative process. Positive impacts include the tool helping composers explore a wider palette of chords, to create chord progressions that were “fresh and different throughout”. It gave them “new ideas of ways to move forward in the piece”, not just on a chord-to-chord level but between several phrases. It encouraged them to make “big jumps

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IUI 2016, March 7–10, 2016, Sonoma, CA, USA.
Copyright © 2016 ACM 978-1-4503-4137-0/16/03 ...\$15.00.
<http://dx.doi.org/10.1145/2856767.2856792>

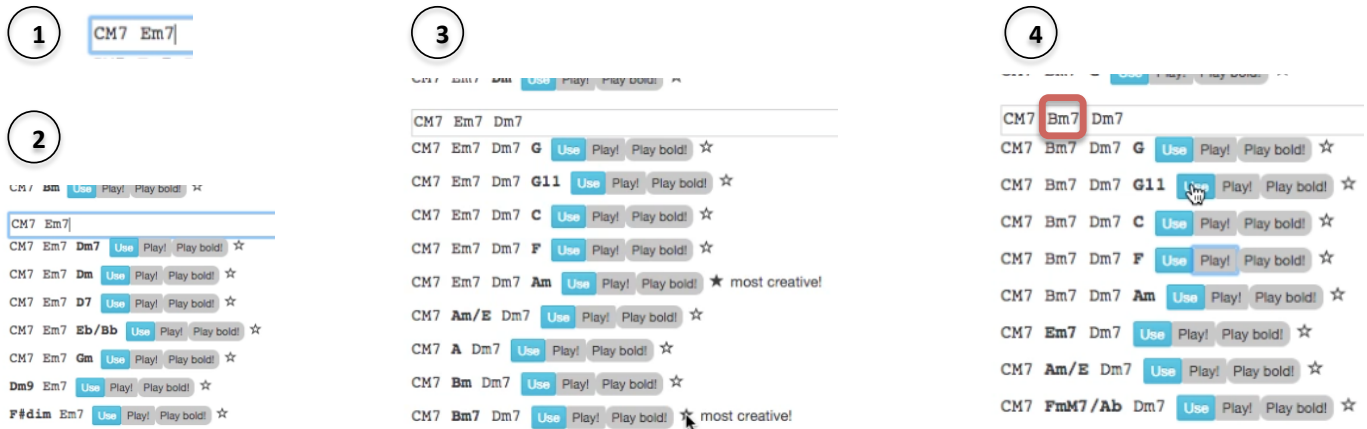


Figure 1. A walkthrough, starting with two chords (step 1), of how CHORDRIPPLE is used to help inspire the next chord (steps 2, 4) and for exploring substitutions (step 3, where the bottom four choices are substitutions of the second chord, which are from the RIPPLE effect of choosing the third chord). As shown, chord recommendations here include both single substitutions and also chords that change the context (what is around a chord change), namely RIPPLES.

in just a few chords”. They sometimes altered their intentions of extending or closing a phrase based on the recommendations they saw. Instead of following the path of least resistance, the recommendations often made them think and work harder. However, some composers found the choices to be too many and that not all chord recommendations were always relevant to what they were trying to achieve. We propose that future systems of this kind should focus recommendations to what is most relevant under the composer’s current intentions. For example, the intention of “make it longer” versus “I’m trying to end here” would entail very different chord recommendations.

As the first study was more exploratory and open-ended, we conducted a second study that was more controlled where students performed transformations on given chord sequences with three design variations of CHORDRIPPLE. We found that the presence of adventurous chord recommendations resulted in students composing more novel chord progressions. Yet we did not observe that recommending more difficult to use chords with RIPPLES allowed students to adopt more of them when compared to just recommending the chords themselves.

The contributions of this paper are threefold. First, we built CR, a creativity support tool that helps composers try more adventurous options; CR supports both adding new material to an existing chord sequence and re-editing existing material (thanks to RIPPLES). Second, we adapted a machine-learning model to learn chord similarity based on actual chord usage. Third, we conducted evaluations that yielded insights into how tools like CR can be integrated into composers’ creative practice.

PRIOR RESEARCH

There have been a number of creativity support tools that assist novice composers in composing chord progressions. Most systems are designed to address the musical problem of harmonizing a given a melody [1, 8, 9, 13].

While these systems all use a data-driven approach, each system strives for a different goal and objective in recommending chords. For instance, [8, 13] allows users to adjust the overall “mood” quality of the accompaniment through knobs such as “happy” which adjusts the weight between the transition matrices of a hidden Markov model for the major and minor key. Users can also modify chords individually, but the recommended chords are restricted to only the most typical chords given the local musical context. [9] provided users with a novel interface with control axes that corresponded to principal components of variances of differential transition dynamics in western popular music, allowing users to more effectively explore a wide range of variations on accompaniment sequences.

[1] takes a hybrid approach of also using music-theory to guide the choice of chords for accompaniment. [4] works with chord sequences by themselves and by performing non-negative matrix factorization on a windowed transition matrix, the system allows users to interpolate between different chord progressions to create their own mix.

CR instead presents a composer with sets of diverse chords that are appropriate but more rarely used in a particular musical context. The support for exploring novel and diverse inspirational examples is critical because they help users to generate more novel and diverse artifacts [6, 10, 12].

CHORDRIPPLE

CR provides two mechanisms to help composers adopt novel chord choices in their compositions. First, CR recommends chord substitutions ranging from typical to adventurous (yet still musically appropriate). Second, because the change of one chord in the middle of a phrase may cause the need to change its context, (a concept we name the RIPPLE effect), the system also includes chord recommendations of varying lengths that change a chord’s surrounding chords. We designed the first mechanism to help composers create raw adventurous

material, while the second mechanism to make it easier for composers to make adventurous revisions to existing material.

Modeling chords for recommendation

To build a system capable of recommending chords, we first need a model of chord transitions and similarities. To represent chords, we use chord symbols that abstract chords into attributes, such as its root, chord type (such as major, minor, diminished), inversion and bass, extensions and alterations, which can amount to many possible unique chords.

Like words in natural language, the distribution of chords follows a Zipf-like distribution. For example, Figure 2 shows chord frequency in the Bach chorale corpus (BACH) in *music21*. A handful of chords are used very frequently, while many chords are rarely used.

Modelling rarely-used chords and chord transitions is difficult. For some applications it is common to reduce the number of chords to 24, treating all chords as either major or minor rooted in one of the 12 pitch classes. However, this comes with the compromise of conflating chords that bear very different transition dynamics. For example, even though $\mathbb{I}64$ is an inversion of \mathbb{I} , it serves a very different function in a cadence. Moreover, as our goal is to make adventurous chord recommendations, we need to be able to model how rare inversions and extensions are used. Hence, we keep all the chord attributes as annotated, which for example results in 92 unique chord symbols in the BACH corpus.

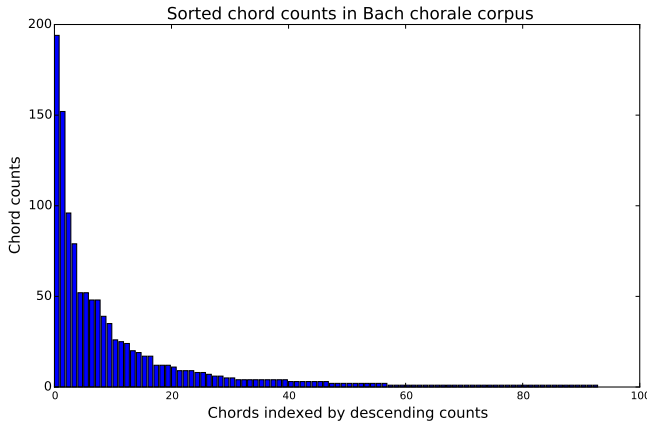


Figure 2. The occurrence count distribution of chords in roman numerals used in Bach chorales as annotated by Tymoczko [15] in the *music21* Bach-chorale corpus [2].

CHORD2VEC

To model chord transitions and similarities, we adapt a neural-network model from natural language processing known as WORD2VEC [7] to the music domain as CHORD2VEC. This model learns chord embeddings from a corpus of chord sequences, placing chords that are used in similar contexts nearby in a vector space. The learned embeddings support substitutions between semantically similar chords. This allows chords that are rarely or never seen in a particular musical context to be recommended if they are nearby in the embedding to other chords that are commonly used in that musical context.

As shown in Figure 3, this model is a skip-gram neural network. Hyperparameter m specifies the dimensionality of the continuous latent embedding, and w specifies the size of the considered context. Latent embeddings of each chord are defined by matrices U , indexed by the input chord, and V , which is indexed by $t+l$, where $-w \leq l \leq w$ indicates output chord c_{t+l} 's relative time position in a sequence with respect to the input chord c_t .

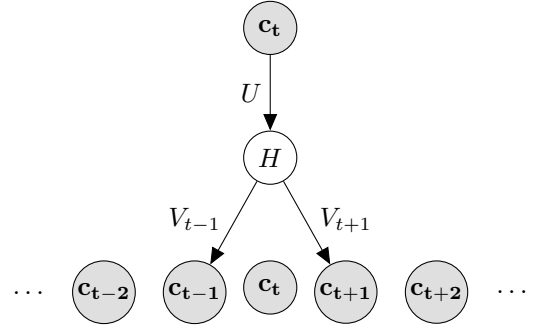


Figure 3. Skipgram neural network with window size $w = 1$.

We use u_i to denote the vector representation of chord i . We use c_t to refer to the chord used at time t , and u_{c_t} and v_{c_t} to refer to the input embedding (U) and output embedding (V) of the chord c_t . Following [7], we define $p(c_{t+l}|c_t)$ as the softmax of the dot product between a chord and its context, as in Eqn. (1), where $-w \leq l \leq w$. The training objective of the model is to maximize the log-likelihood of all chord sequences independently, as in Eqn. (2). Maximizing this objective has the side-effect of putting chords used in similar contexts close to each other in the input embedding U .

$$P(c_{t+l}|c_t) = \frac{\exp(v_{c_{t+l}}^T u_{c_t})}{\sum_{i=1}^{|W|} \exp(v_i^T u_{c_t})} \quad (1)$$

$$L(c_0 \dots c_{T-1}) = \sum_{t=1}^T \sum_{-w \leq l < w, l \neq 0} \log P(c_{t+l}|c_t) \quad (2)$$

Trained embedding for recommendation

To support CHORDRIPPLE, we trained CHORD2VEC on a corpus of 200 Rock songs (all transposed to \mathbb{C}) from the *Rolling Stone Top 500 Hits* [14] containing 98 distinct chords, using latent dimension $m = 10$ and window size $w = 1$.

Data and exploratory experiments

In our experiments, we use the two aforementioned annotated corpora, *Bach Chorales* from *music21* [2] annotated by Tymoczko [15] and *The Rolling Stone: Top 200* corpus transcribed and annotated by Temperley and Clercq [14]. To model the rich palette of chord alterations, we retain all chord attributes from the original chord annotations. We experiment with both transposing chords to the key of \mathbb{C} versus preserving the chords in their original keys. By transposing we assume that key does not have a major influence on how chords are used. By keeping songs in their original key one can also model key-specific trends at the expense of having less counts on each chord.

Visualizing the axes of the latent embedding

To gain more intuition on how the latent embeddings capture musical semantics, we trained a toy bigram version of CHORD2VEC with hidden layer size $m = 1$ for ease of interpretation. We reduced the number of chord symbols by requiring that they have at least 5 occurrences, which left 30 chords. We can see that these 1-D embeddings approximately capture functional harmony. In Figure 4, we see that chords close in U share similar target chords, for example, V and $viio$ chords and its inversions and seventh chords are closely related, as they are often followed by some form of tonic chords. In Figure 5, chords close in V serve as similar target chords, for example as a phrase moves into a cadence, chords that transition to a $I64$ are likely to choose $ii7$ as an alternative or directly move to a V chord.

Visualizing embedding when trained in original keys

When training CHORD2VEC on the ROCK corpus with songs kept in their original keys, the learned embedding exhibits topological properties that correspond to musical relationships. The model had a hidden layer size of $m = 10$ and window size of $w = 1$. The x and y axes in Figure 6 correspond respectively to the first two principal components of the projection layer U . It shows that the major and minor chords are both arranged in the latent space in shapes corresponding to the circle-of-fifths. One interpretation of this result is that the tonic and dominant tension relationships are relatively strong in songs of all keys, causing them to push each other apart, some stronger than others causing the angles of the polygon to be different.

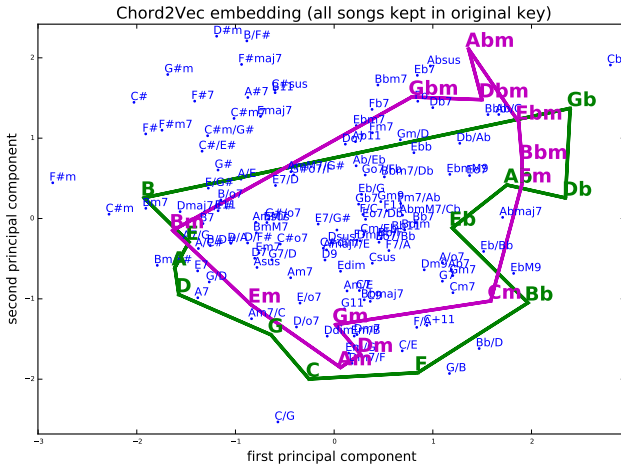


Figure 6. The CHORD2VEC projection layer U with hidden layer size $m = 10$ projected onto its first two principal components. The model was trained on the ROCK corpus with songs kept in original key, and then. Major and minor chords are both arranged in the latent space in shapes corresponding to the circle-of-fifths.

System Design

CR recommends a mix of adventurous and commonly-used chords based on the current context. To recommend more adventurous chords, we use the latent embedding learned from CHORD2VEC to query chords that are the most similar to a composer’s current chord choice.

Following [7], we define the similarity between chords through the cosine distance of their embeddings vectors, as in Eqn. (3). Hence, given a chord c_i , we can query the embedding for the k closest chords to c_i to recommend to users as substitutions, by starting with Eqn. (4).

$$\text{similarity}(c_i, c_j) = \frac{u_{c_j}^T u_{c_i}}{\|u_{c_j}\| \|u_{c_i}\|} \quad (3)$$

$$\text{mostSimilar}(c_i) = \arg \max_{c_j} \text{similarity}(c_i, c_j) \quad (4)$$

These queried chords are not necessarily the most typical in the current musical context as preference has been put on being similar to a composer’s current chord choice. To recommend what is typical given a musical context, we learn a distribution $P(c_t|c_{t-1})$ using a simple bigram model. WORD2VEC also learns a similar probabilistic model, but conditioned on both c_{t-1} and c_{t+1} . For simplicity in the current implementation we adopt a bigram model for these queries.

RIPPLES

We call recommendations that affect more than one chord RIPPLES. A RIPPLE can help smooth out transitions between chords, making it potentially easier for users to swap in more adventurous chords. It can also give new ideas of how to move forward or backward from a chord. The basic RIPPLE recommends both a substitution for the current chord (c_t) and also a reworking of its context. There are two steps involved in generating such RIPPLES. First, the substitution (c_t^s) is generated either by querying the CHORD2VEC embedding or by inward conditioning on the current context, in this case (c_{t-1}, c_{t+1}). Next, the immediate current context (c_{t-1}, c_{t+1}) can be reworked by conditioning on its larger outer context and the newly substituted chord, (c_{t-2}, c_t^s, c_{t+2}).

We can derive different kinds of ripples through a cascade of inward and outward conditionals. For example, one can imagine regenerating the larger context $\{c_{t-2}, c_{t+2}\}$ by conditioning outward from the newly substituted context (c_{t-1}^s, c_{t+1}^s). For each of the conditioning and substitution steps, we can also imagine sampling them to generate more variations.

We initially ranked the chord recommendations by the transition likelihood (Eqn. (2)) of the new sequence, using dynamic programming under the bigram model. Table 1 gives an example of chord recommendations from CR when given the most basic chord progression, $C F G C$. We clearly see that the recommendations from a purely NGRAM model are much more conservative, consisting of mostly C and F chords, while WORD2VEC is much more adventurous, recommending chords that can help composers break out of commonly used chords.

However, if we only consider how well the ripple recommendations flow in and out of the substitution chord, we will be neglecting the original structure of the sequence. For example, in Table 2, when regenerating the context for the chord F in the sequence $C F Dm G C$, the smoothest path to transition into its substitutions F or F/A is to precede and follow it with the chord C . The result is not very exciting and by returning to C it breaks down the phrase into two sub-phrases.

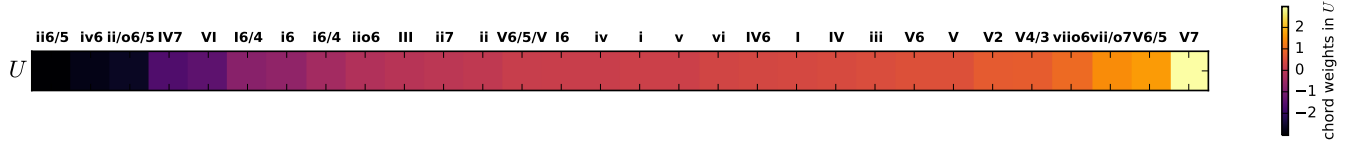


Figure 4. The inputs weights U of the bigram CHORD2VEC on the Bach chorales, with latent embedding size $m = 1$. Some predominant and subdominant chords cluster on the left while all the dominant chords cluster on the right.

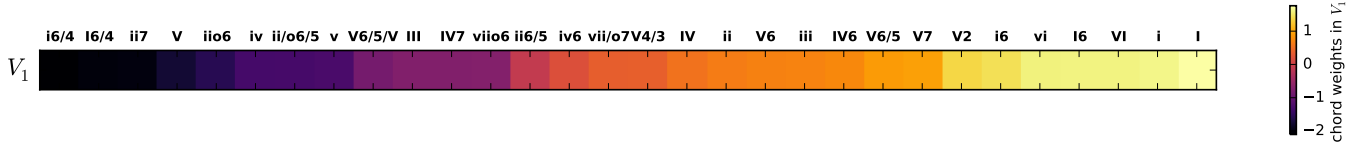


Figure 5. The output weights V of the bigram CHORD2VEC on the Bach chorales, with latent embedding size $m = 1$. Tonic-like chords cluster on the left end while mostly predominant and subdominant chords on the right

C F G C	single sub, <i>above</i>	triplet subs, <i>above</i>	outward subs, <i>below</i>
N-GRAM	C, F	C F C F, C C F C	C C G F, C Am G Am
WORD2VEC	Ab7, Dm7	C F Ab7 Cm, C C Dm G	C F7 G B, C F/A G C7

Table 1. Comparison between chord recommendations from N-GRAM and WORD2VEC the G chord, third in the chord sequence C F G C. The chords in bold function as substitutions (sub) for the original sequence. The labels *above* and *below* indicate where these recommendations would appear relative to the input textbox in CR.

To preserve the original structure or add meaning variation to that structure, we need to also consider the similarity of the chords in the regenerated context to that of the original. This can be achieved by maximizing both the transitional likelihood between the substitution chord, it’s regenerated immediate context and the larger original context and also the distance between the immediate context to the substitution chord in the regenerated sequence and the original sequence, as shown in Eqn. (5). We denote the new sequence with substitutions as \mathbf{C}^s and the original sequence as \mathbf{C} , and use k_1 and k_2 to denote the start and end index offsets for the sequence of substitution. λ controls the balance between smooth transition and similarity to original sequence. This is the RIPPLE recommendation implemented in the second version of CHORDRIPPLE where we used $\lambda = 1$.

$$\text{cost}(\mathbf{C}^s, \mathbf{C}) = -L(\mathbf{C}^s) + \lambda \cdot \text{sim}(\mathbf{C}^s, \mathbf{C}) \quad (5)$$

$$\text{sim}(\mathbf{C}^s, \mathbf{C}) = \sum_{-k_1 < l < k_2} \log \text{similarity}(c_{t+l}^s, c_{t+l}) \quad (6)$$

The example in Table 2 shows the advantages of this additional constraint. Not only does it regenerate the context with chords that respect more the original function of the chord, it also regenerates the context in a way that is stylistically consistent with the substitution chord (using a seventh chord C7 to lead into F7).

User interface

To make it easier for composers to experiment with these recommendations, we designed a user interface that allows composers to see alternatives in relation to their own choices, akin to autocompletions. As users type their chord sequences

Ripple method	Generated sequence
Transition likelihood only	C F/A C G C C F7 C G C
Similarity to context	Cm F/A Ab G C C7 F7 D G C
Original sequence	C F Dm G C

Table 2. Comparison between ripple recommendations that only take transition likelihood into consideration when regenerating the context, versus recommendations that also consider the similarity of the regenerated context to the original

into the input text box shown in Figure 1, suggested chord sequences appear in parallel both above and below, with chord changes highlighted in bold. Users can listen to how their chord sequence might sound if they had chosen a particular suggested sequence. More typical chord recommendations are near the text box, while more adventurous ones are at the fringes. Composers can use the recommendations to help select the next chord, or they can go back to earlier chords and see the bolded changes ripple backwards.

As shown in Figure 1, chord sequences below the text box are recommendations for what can be changed around the current chord. For example, the first five rows below show the top five typical continuations. The bottom two rows show how the chord before the most recent chord be approached differently, which in step 2 of Figure 1 is the second chord in the sequence. Cropped from above the text box are pairs of recommendations that include a single substitution for the current chord and triple substitutions for the chord and its contexts. The top half are the adventurous ones and the bottom half the typical. By clicking on the USE button, users can easily replace their

own chord sequence with one of the recommended. A walk-through of Figure 1 is given in the next section.

EVALUATION

We conducted two user studies in order to understand how CR impacts the compositional process, each with different degrees of emphasis on external and internal validity. First, we wanted to understand how music students would relate to the tool, and in what ways the tool can assist them in composing chord progressions. We also wanted to observe if the tool can help students go beyond their usual palette of chords. In this exploratory study, we asked students to use the tool to help them create variations of chord progressions in their own music. They were hence more intrinsically motivated. For the second study, we conducted a controlled experiment to specifically study the effects of adventurous and RIPPLE recommendations respectively on the novelty of students' chord progressions and their satisfaction. Each student was asked to perform transformations on given chord sequences with three design variations of CR.

EXPERIMENT 1: STRUCTURED OBSERVATIONS

We want to understand how CR impacts the compositional process. What kind of goals and expectations do composers have when working with the tool? How does the tool support or alter their intentions? How do they choose between different chord recommendations? To answer these questions, we took a structured-observation approach [5] to our studies. We recruited 15 participants to use CR to further develop the chord progressions they were working on for a piece. We logged their edit traces. We interviewed the participants both before and after the composing task. For the last ten participants, we audio-recorded and transcribed their interviews, and screen captured their interactions with CR.

Participants

We recruited 14 undergraduate music students and a graduate student in the applied sciences with an interest in music. The total of 15 participants included five females and ten males. We paid them \$10 each for a 55-minute study session. Fourteen participants were undergraduate music students from three local universities. All the music students had some composing experience, ranging from composing for harmonization assignments in music theory classes to writing their own symphonic pieces. Most students played either the guitar or the piano, and used the instrument as a means to composing chord sequences. Most students composed roughly in the genres of pop, rock and jazz, or with traditional classical harmonies.

The composition tasks

We asked each participant to bring in a piece that they were currently working on that involved chord progressions. We suggested three possible tasks for using our tool: (1) to create variations of their existing chord sequences, (2) to use a few chords from their chord sequences as a seed to growing a new sequence, (3) to compose a new chord sequence from scratch. These tasks were structured enough to provide us with some degree of uniformity for comparison and generalization across participants, but still open enough to support individual expression.

Procedure

Each participant was asked to bring their scores and sketches of a current piece they were working on, with the chord progressions they used in guitar-tablature-like notation. Each session began with a short semi-structured interview on their background in music, how they compose, the role of chords in their music. They were then asked to walk-through how they composed the piece that they brought in. After the initial questions, participants were introduced to CR. To familiarize them with the chord notation used by the interface, participants were asked to type in the chords sequences that they brought in. We then suggested to them three possible short compositional tasks to perform with CR, which we will further outline below. Depending on how much time a participant took to complete the first task, each participant typically completed one to two of the tasks. During the task, participants were asked to adopt a think-aloud protocol. To further understand why and how participants were making decisions at particular moments, we interjected with probing questions. After each task, we conducted a short semi-structured interview for retrospective verbal accounts of how they felt the tool had impacted their creative process.

Two detailed walkthrough of CR usage

We start by walking through a long and short example to illustrate how composers use the tool, how they make decisions, and the impact of the tool on their creative process and the creative outcome.

Composing a second phrase as a variation

Participant 7 (P7) came in with the following chord progression for a minimalistic piece, where this sequence would be looped: C Am Dm F C+. He wanted to explore what else he could do with it. Before he started, he decided he wanted to keep the sound of the augmented chord C+ at the end to preserve the identity of the phrase, but “break out of the really diatonic stuff in the middle”. When seeking substitutions for the second chord, he manually replaced Am with Em, which are both possible tonic substitutions that can function as tonic prolongations of C. In this case the composer explored closely related chords, while the tool suggested a more unusual alternative Bb.

C Bb Dm F C+: “This one is not diatonic, [it makes] an aggressive shift, which I like. I feel like that’s something that could be used as a climatic sudden moment not so much as a thing that can be repeated”.

Even though he liked the effect of Bb, it was not appropriate for the current context as the chord sequence at stake was supposed to be looped, but the suggested chord was meant to be a one-time “sudden surprise change”. He then reverted back to the second chord being Am. Figure 7 shows the alternatives recommended by the tool. The chord recommendations Fmaj7/C and Fmaj7 on the top two rows are the chords that are most similar to Am according to the WORD2VEC model. In this case, these chords are also mutually similar, being inversions of each other. The bottom two chords, Em and F, are the most likely chords given the current surrounding context being C and Dm. In roman numerals, they would be I iii ii and I IV ii, the latter being more common than the

former. The composer eventually chose Fmaj7/C, which holds the bass note of the previous chord C. He liked that the chord was an inversion and that it was a major 7 chord, saying “It’s a good sound. It’s different from what I am using”.

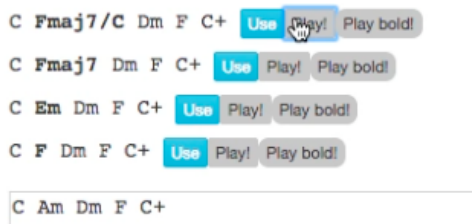


Figure 7. Chord recommendations for Am, the second chord in the sequence C Am Dm F C+. The top two rows are SIM type recommendations, while the bottom two rows are IN-RIPPLES.

At the end, the composer came up with two new phrases for his piece, C Fmaj7/C Gm7 F C+ and C Fmaj7/C Bb C Cm G. For the second to the last chord, he chose Cm because he realized it links to two phrases with an interesting internal logic, that is, C “opened” up to go to C+ in the first phrase, while in the second phrase C went in the opposite direction and “contracted” to Cm. The tool gave him “ideas of ways to move forward in the piece, previously I just had this as a repeating progression”. He felt the chord sequences were consistent with his style in a way that they bore “grains of progression” that [he] uses a lot, but at the same time the tool helped him go beyond the ordinary. The tool helped him to explore the in-between space:

“If I am trying to come up with something tonal but [with] variation ... I have difficulty. It’s either really tonal or its not and I am not good at the in between, sort of like tonal but breaks free, so that’s I think what I accomplished here a little bit”.

Using RIPPLES to change what was fixed

Participant 8 (P8) came in with a 12-chord long sequence. He had come up with the first two chords, CM7 Em7, on the guitar and he “felt like [he] could go somewhere with it” and that he could “really make that build”. But when he went to the piano, it took “a different direction”.

For the study, we decided to start with the same two chords, CM7 Em7, and see what other directions the tool would take it to. The composer initially worked forward, exploring continuations (Figure 1 STEP 2), and then chose Dm7 as the third chord. In Figure 1 STEP 3, he continued to work forward, and marked the most untypical continuation as the MOST CREATIVE. But upon seeing that there were also recommendations for how to change the existing chords (bottom four rows), he chose CM7 Bm7 Dm7, which replaced the original second chord Em7 with Bm7 (result shown in Figure 1 STEP 4). This recommendation was based on its surrounding chords CM7 and Dm7. Note that Dm7 was a chord inspired by the tool earlier.

Even though the composer originally thought of the first two chords as fixed, he was flexible and open to changing them when he saw a substitution he liked. This example also illustrates how the tool could inspire a cascade of changes.

Overall expectations and attitudes towards CR

Participants’ reactions to CR were mixed. We give a few brief examples of how participants reacted to CR for the different tasks. For the task of seeking variation to an existing chord sequence, we saw two contrasting intentions from composers. For example, P14 was open to new ideas of how the sequence could be varied: “I was more open to changing up the direction of it or the feeling of it”. Meanwhile, for P7, it was the opposite. He had a more specific idea about what the desired variations would sound like and how they would relate back to the original sequence: “[I am] more picky about how I am changing it. [It] took longer [than composing from scratch] to go through [the chord recommendations] and find the right chord”.

For the task of starting from scratch, we also saw two contrasting experiences. For example, P7 felt that composing with CR was much faster than improvising on a piano: “I like it when I’m composing from scratch. I have less an idea of what I want, hearing what’s next, makes the process quicker”. In contrast, P15 found it difficult to generate a chord sequence from scratch, partly because this way of working was foreign to him. He usually writes songs by collaborating with his friend who would first compose a melody and pass it on to him, and he then would come up with chords to accompany it.

Choosing between chord recommendations in CR

There are often several different levels of decisions at play when a composer is trying to decide which chord to choose. For example, [3] outlined four decision levels in a compositional process, which consists of the objective, the musical idea, grammaticality and technical aspects. Moreover, a composer’s decision-making criteria are dependent on the stage she is in in the compositional process and also where she is structurally in the piece. Furthermore, a composer might be trying specifically to find the chords she is hearing in her mind [P9] or be open to new ideas [P14].

The different decision levels are often interconnected. A composer may have an overall objective of “breaking out of diatonic harmony” [P7]. Similarly, P14 says “I’m looking more for less conventional chords”. These aesthetic aspirations can impact what composers consider as grammatically plausible. On the other hand, a certain chord may work well grammatically but “changes feel” [P14], changes what the phrase is expressing. This elevates the level of decision to that of the musical idea.

When choosing between recommendations in CR, composers often anticipate what subsequent changes or continuations may be needed. For example, choosing one chord may lead to the need to change the next two chords [P14] because it does not flow well into those two chords which are themselves a unit. P1 struggled between a chord she liked more but would require more work to complete the phrase, versus a chord that would conclude the phrase immediately.

Impact of CR on the composition process

Upon first encountering the tool, many participants felt compelled to listen to all of the recommendations. P14 commented “I want to listen to all of them, even if it might not make sense

just looking at it, because why not, maybe it will spark something”. P15 felt the options were too many and that it was hard to choose between them. It was “like too many ice-cream flavours when you only want three”. In contrast, P9 said “I use this like an inspiration and so seeing, having it suggest a ton of stuff to me, I was just like getting more information faster”.

Participants had different interpretations on the nature of the recommendations and their effects. P9 described the tool as “yelling at you with things you won’t think about”, and when he “sees something crazy, try it, get more crazy”. This sentiment was echoed in P6. Upon seeing a chord recommendation that increased tension, she was inspired to want even “more intensity, so [she] added a 7th”. P12 said the tool helped him “see where it can go from here”. P9 likened the tool to a co-pilot that gives him validation when he sees the chords he wants to try out among the suggested chords. Many composers were delighted to explore a wider palette of out-of-key, extended, altered and inverted chords. P7 said the tool helped him stay “out of straightforward diatonic space”, and that it suggested “new places to go”. While others found some of the out-of-key chords to be irrelevant: “It doesn’t go together. I just feel like it’s random” [P14].

Let us zoom into out-of-key chords for a moment. Composers find it harder to workout out-of-key chords manually: “It is hard to generate these chords without hearing how they sound first” [P8]. CR helped composers imagine and hear how to transition to and from and between out-of-key chords, making them more accessible.

“That’s a big jump for me to make in just a few chords, something I’m not able to do on my own, play around and land there” [P7].

P7 pointed out that CR helped him infer which key he was in by suggesting continuations that showed him “this is where you are going” [P7]. He said he would otherwise have forced the sequence to end in the wrong key and destroyed it: “it saved me from myself”, said P7.

P8 described the experience of working with CR as faster than working things out in his head, while P6 felt it was a much faster process than improvising on a piano:

“It is just faster, instead of doing that in my head, these are all the options that make sense, just try them and see if one of these are what you are thinking about” [P8].

The tool at times altered the workflow of the composer by inspiring new directions. For example, P7 originally planned to fix the beginning and the end of the phrase and just work on the middle. But upon seeing the chord recommendations for the middle chords, he chose chords that suggested new directions which required the changing of the ending chords.

EXPERIMENT 2: CONTROLLED EXPERIMENT

In contrast to the first study, we designed a controlled experiment to measure how the level of adventurousness of chord recommendations and how RIPPLES respectively impact the novelty of students’ chord progressions and their level of satisfaction.

Task, procedure, participants

The task was to transform a given 8-chord long progression to reflect the character and mood of an image, such as Van Gogh’s *Starry night*. The participants were told that they would not be evaluated on how well the chord progression reflects the image, but that they can see the image as a source of inspiration.

The independent variable of this experiment was the recommendation type. We designed three variations of CHORDRIPPLE: for the current active chord, recommends

- SINGLETON-TYPICAL: only single-chord substitutions that are the most typical given the context
- SINGLETON-ADVENTUROUS: only single-chord substitutions that are similar to current active chord but less typical in the given context
- RIPPLE: single-chord substitutions as in SINGLETON-ADVENTUROUS and also RIPPLE substitutions that include them as the middle chord.

Each participant was first asked to complete a tutorial that involved trying out all the functionality of CHORDRIPPLE. After completing the tutorial, each participant was asked to complete all three conditions. The order of the conditions was randomized. Each condition was paired with a different initial chord progression and image.

Similar to the first set of studies, we recruited 11 music student from local universities. One student had to leave early for class and did not complete the study. The system crashed for another student in the middle of a condition. At the end, we collected 9 complete logs.

Hypotheses

We test two hypotheses regarding how recommendation type might have an impact on the novelty of chord sequences composed and students’ satisfaction.

- H1: SINGLETON-ATYPICAL will result in more novel sequences (and higher satisfaction) than SINGLETON-TYPICAL
- H2: RIPPLE will result in more novel sequences (and higher satisfaction) than SINGLETON-ATYPICAL.

Design and analysis

We used a within-subjects design with one factor (recommendation type) with three levels (SINGLETON-TYPICAL, SINGLETON-ATYPICAL, and RIPPLE). We measured two variables: the user’s satisfaction with the chord progressions they composed on a 5-point Likert scale, and also the novelty of their top-rated chord progression. We define NOVELTY of a chord progression $c_{0..T-1}$ as the average of the inverse counts (as measured in the ROCK corpus) of the constituent chords, as shown in Eqn. (7).

$$\text{novelty}(c_{0..T-1}) = \sum_{i=0}^{T-1} \frac{1}{N_{c_i}} \quad (7)$$

We used the nonparametric Friedman test to first check if there are differences among the different levels. If the test is significant, we then perform the planned pairwise comparisons following the hypotheses and use the nonparametric Wilcoxon signed-rank test to see if they are statistically significant.

Main effects and discussion

There was no effect of the recommendation type on the satisfaction of the student's ratings on their best chord progression. The means and standard deviations of the ratings in each of the conditions are shown in Table 3. Note that the statistical tests computed are within-subjects and the table is just for giving an overview on the distributions of students' ratings.

We did, however observe a main effect in recommendation type on the novelty of students' best-rated chord progressions. The means and standard deviations of the NOVELTY scores in each of the conditions are shown in Table 3, again just to show the overall differences across conditions. The Friedman's test shows that there are differences in novelty among the three conditions within subjects, and that the effect is significant ($\chi^2(2, N = 9) = 12.67, p = 0.0018$). We found that when composing with the SINGLETON-ATYPICAL design variation of CR, students generated chord progressions more novel than when using the SINGLETON-TYPICAL design. The Wilcoxon signed-rank test shows that this effect was statistically significant ($p = 0.0039$), as shown in Table 4. Hence, this supports the novelty aspect of H1.

We did not observe a statistically significant effect for the pair RIPPLE between SINGLETON-ATYPICAL ($p = 0.0977$). Hence, H2 is not supported. We also performed a post-hoc pairwise comparison and found that the RIPPLE condition resulted in progressions that were more novel than the SINGLETON-TYPICAL and the effect was statistically significant ($p = 0.0273$).

These results show that recommending more adventurous chords can help students compose more novel chord progressions. In particular, our chord embedding model was able to make chord recommendations that help students go beyond the ordinary. We did not observe that RIPPLES made it easier for students to incorporate more adventurous chords. We speculate that it is because RIPPLES tend to smooth things out and use less atypical chords in a row. Moreover, novelty itself does not always lead to satisfying progressions. For example, a chord progression may contain many novel chords but incoherent.

Furthermore, there are also many reasons why a RIPPLE might be chosen, beyond smoothing out the transitions between the original context and recommended novel chord. For example, a RIPPLE may be used because it is an interesting variation of the original context, or because the RIPPLE changes more chords at once and allows users to more quickly explore different ideas.

BROADER DISCUSSION AND DESIGN IMPLICATIONS

We have observed that CR had many positive effects on composers' compositional process, although the particular outcomes depended on the person and the details of the task they

were working on. Therefore, while this approach is overall helpful, a one-size-fits-all solution is not optimal.

There are several factors that affect how adventurous the composer may want the chord recommendations to be. First, a composer's musical and stylistic aspirations would influence their receptiveness to less frequently heard chords. For example, P3 described his goal as writing a "poppy" song that would be easy to relate to, and wanted to use chords that the general audience would find familiar. P5 was seeking chords that carried the sound typical of a genre. Many other students had the goal of "breaking out", and wanted to use chords that would be less likely encountered if they were improvising on their instruments.

Also, adventurousness is contextual. P8 considered using a triad among a chord sequence of mostly seventh chords as a bold move, as the triad really "stands out". At other times, the intention of the composer may be to find "something simple" so that it does not take away from the other more "iconic moments" of the phrase [P8].

We specifically designed CR to provide suggestions ranging from very typical to very adventurous, so that it could accommodate the diverse needs of composers. However, CR currently takes a fixed approach, splitting its recommendations to be half typical and half adventurous, but composers at times wanted more of one kind. Composers sometimes found some of the recommendations to be too similar to each other.

We propose that designers could include a mechanism that explicitly allows composers to adjust how typical to adventurous they want the recommendations to be. Systems should also allow composers to control the scope of diversity for recommendations, so that they could adjust between widely exploring the chord palette versus zooming into the variations of a particular sound. Furthermore, composers have different goals for different parts of a piece. For example, the intention of "making it longer" versus "I'm trying to end here" would entail very different chord recommendations. The former may include chords that open up new directions while the latter would focus on chords that provide a resolution. We propose that there could be a "knob" that allows users to control the amount of "tension" desired in the chord recommendations. It may also be possible to automatically infer a composer's intention for "conclusiveness" based on the musical context and the chords she has explored so far. The advantage of a knob is that it affords the user with more fine-grained control and the potential to explore a wider range of possibilities iteratively. The advantage of inferred intentions is that they can be used to automatically adapt the chord recommendations, allowing composers to see only what is relevant and thus making the process of composing faster.

ACKNOWLEDGEMENT

I had countless discussions with Ken Arnold throughout the duration of this work. Hubert Ho helped me tirelessly in all aspects of logistics. I want to thank MuseAmi for their feedback and support during summer 2015, especially Eric Humphrey and Leigh Smith at MuseAmi Labs.

Measure	SINGLETONTYPICAL	SINGLETONATYPICAL	RIPPLE
Self-rated satisfaction	4.5 (0.707)	4.06 (0.682)	4.222 (0.939)
Novelty	0.021 (0.035)	0.147 (0.091)	0.063 (0.085)

Table 3. The mean (standard deviation) of the two measures, self-rated satisfaction and NOVELTY, of student’s best chord sequence in the three conditions.

SINGLETONTYPICAL – SINGLETONATYPICAL	SINGLETONTYPICAL – RIPPLE	SINGLEATYPICAL – RIPPLE
$S = -22.50, p = 0.0039$	$S = -18.50, p = 0.0273$	$S = 14.50, p = 0.0977$

Table 4. Comparisons on the novelty of student’s best chord sequence between all the pairs of conditions, reporting the test statistic S and two-tailed significance p from the the Wilcoxon signed-rank test.

REFERENCES

1. Chuan, C.-H., and Chew, E. A hybrid system for automatic generation of style-specific accompaniment. *Proceedings International Joint Workshop on Computational Creativity*. (2007).
2. Cuthbert, M. S., and Ariza, C. music21: A toolkit for computer-aided musicology and symbolic music data.
3. Delalande, F. Towards an Analysis of Compositional Strategies. *Circuit: Musiques contemporaines 17*, 1 (2007).
4. Fukayama, S., Yoshii, K., and Goto, M. Chord-sequence-factory: A chord arrangement system modifying factorized chord sequence probabilities. *International Society for Music Information Retrieval Conference ISMIR* (2013).
5. Garcia, J., Tsandilas, T., Agon, C., and Mackay, W. E. Structured observation with polyphony: a multifaceted tool for studying music composition. In *DIS '14: Conference on Designing Interactive Systems, DIS '14*, ACM (2014), 199–208.
6. Marsh, R., Landau, J., and Hicks, J. How examples may (and may not) constrain creativity. *Memory and Cognition 24*, 5 (1996), 669–680.
7. Mikolov, T., Sutskever, I., and Chen, K. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*. (2013), 1–9.
8. Morris, D., Simon, I., and Basu, S. Exposing parameters of a trained dynamic model for interactive music creation. *Proc AAAI 2008* (2008), 784–791.
9. Nichols, E., Morris, D., and Basu, S. Data-driven exploration of musical chord sequences. *Proceedings of the 13th international conference on Intelligent user interfaces IUI 09* (2009), 227.
10. Nijstad, B. A., Stroebe, W., and Lodewijkx, H. F. Cognitive stimulation and interference in groups: Exposure effects in an idea generation task. *Journal of Experimental Social Psychology 38*, 6 (2002), 535 – 544.
11. Serrà, J., Corral, Á., Boguñá, M., Haro, M., and Arcos, J. L. Measuring the evolution of contemporary western popular music. *Scientific reports 2* (2012).
12. Siangliulue, P., Arnold, K. C., Gajos, K. Z., and Dow, S. P. Toward collaborative ideation at scale: Leveraging ideas from others to generate more creative and diverse ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW '15*, ACM (New York, NY, USA, 2015), 937–945.
13. Simon, I., Morris, D., and Basu, S. MySong: automatic accompaniment generation for vocal melodies. In *Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems*, vol. 1 (2008), 725–734.
14. Temperley, D., and Clercq, T. D. Statistical Analysis of Harmony and Melody in Rock Music. *Journal of New Music Research 42*, 3 (Sept. 2013), 187–204.
15. Tymoczko, D. Local Harmonic Grammar in Western Classical Music. Available at <http://www.dmitri.mycpanel.princeton.edu/mozart.pdf>.