Harmonic Analysis of Music With Combinatory Categorial Grammar

Second-Year Report

Mark Granroth-Wilding

October 6, 2011

Contents

Chapter 1

Introduction

Hierarchical structure can be identified in rhythmic patterns of musical melodies and the harmonic progressions that underly them. Similar structure is found in the prosody and syntax of language, commonly analysed using trees diagrams that divide a passage of speech or text recursively into its constituents, down to the level of individual words. It is natural to wonder if the techniques used to process natural language can be applied to the interpretation of music.

In natural language processing (NLP), analyzing the syntactic structure of a sentence is often a prerequisite to semantic interpretation. The main obstacle to such analysis is the high degree of ambiguity in even moderately long sentences. In music, a similar sort of structural analysis over a sequence of notes is fundamental to tasks such as key identification and score transcription. These tasks in general depend on both a harmonic (tonal) analysis and a rhythmic (metrical) analysis.

If harmony contains syntactic structure like natural language, then perhaps it is possible, as it is in the study of languages, to express this structure using formal generative grammars. Many have speculated so before us and certain authors have even attempted to write grammars of musical harmony and of other structured aspects of music, such as melody (Keiler, 1981; Lerdahl & Jackendoff, 1983; Winograd, 1968; Lindblom & Sundberg, 1969). Before we can justify the formulation of any such grammar, we must ask what type of semantic analysis the syntactic structures exist to produce. Any finite sample of a language's stringset can be modelled using a regular grammar, the least expressive level in Chomsky's hierarchy. We can only answer the question of how expressive a grammar formalism is required to represent a particular language by reference to the semantics of the language. Any attempt to reason about the expressive power required of a grammar purely on the basis of the strings of a language in fact relies on intuitions regarding the language's semantics and the syntactic structures that serve the map the strings onto that semantics (Steedman, 2000).

We propose a level of harmonic analysis of Western tonal music based on a formal theory of tonality which we treat as analogous to a compositional semantics of natural language. We present a variant of the grammatical formalism of Combinatory Categorial Grammar designed to handle the sort of syntactic structures needed to perform this semantic analysis. Using this formalism, we construct a lexicon for a grammar to produce analyses of chord sequences. In NLP, probabilistic models based on statistics gathered from linguistic cor-

pora are used to inform and speed up the automatic process of interpretation through parsing. Using a corpus of jazz chord sequences, we carry statistical parsing techniques from NLP over to our problem of musical parsing.

Most of our work so far focusses on the problem of performing automatic harmonic analysis of chord sequences. A harder problem is that of performing the same sort of analysis of sequences of notes. We have begun to address this problem by examining another approach, that of Raphael & Stoddard (2004), to a related problem and considering how their approach may be adapted to our task. We present here some initial experiments to gain insight into the performance of the Raphael and Stoddard model and some thoughts on how a model for our analysis task might be constructed.

One example of a task that the resulting model could be applied to comes from the field of music information retreival (MIR). The harmonic analysis should provide a better signal to compare performances and identify variants on the same song (contrafacts) than the notes themselves. We plan to use our grammatical harmonic analysis model for this task and compare its performance to other approaches.

In chapter 2, we give an account of the problem of harmonic analysis and the background to our approach, then describe the grammar formalism we use and the lexicon for our grammar of jazz chord sequences. We then present some experimental results comparing the performance of several statistical parsing techniques. This chapter is a draft of a journal paper that has not yet been submitted for publication.

Chapter 3 gives a more detailed description of the way we formulate the semantics of tonality and how this is used by the grammar presented in chapter 2, which skated over the precise details of the semantics.

In chapter 4 we present our work so far based on the model of Raphael & Stoddard (2004). The adaptation of this model to our purposes is a work in progress and we present some thoughts on an approach to this task that we intend to explore.

Harmonic analysis is an important aspect of understanding music and is a prerequisite for many musical tasks. We describe formal harmonic analysis in terms of the tonal space of Christopher Longuet-Higgins and a grammar, with statistical modeling techniques, that allows us to perform this analysis automatically given a chord sequence. We show that this grammatical model outperforms a closely related shallower statistical model which does not use a grammar. We propose a simple approach to combining both these approaches to produce a robust method for automatic harmonic analysis of chord sequences.

The analysis and much of the modeling involved are not restricted to use on chord sequences, can apply to a score, or the notes of a performance. Ultimately, our aim is to perform automatic analysis on raw streams of notes and to apply the models to a MIR task. We report some analysis of a related model, that of Raphael & Stoddard (2004), which performs a different sort of harmonic analysis of symbolic note signals. We propose a model using some of the insights of this model to apply our grammatical analysis to note streams.

Chapter 2

A Robust Parser-Interpreter for Jazz Chord Sequences

This chapter is a slightly modified form of an unpublished draft of a paper by Mark Steedman and Mark Granroth-Wilding.

2.1 Introduction

Hierarchical structure can be identified in rhythmic patterns of musical melodies and the harmonic progressions that underly them. Similar structure is found in the prosody and syntax of language, commonly analysed using tree diagrams that divide a passage of speach or text recursively into its constituents, down to the level of individual words. It is natural to wonder if the techniques used to process natural language can be applied to the interpretation of music.

In natural language processing, analyzing the syntactic structure of a sentence is often a prerequisite to semantic interpretation. The main obstacle to such analysis is the high degree of ambiguity in even moderately long sentences. In music, a similar sort of structural analysis over a sequence of notes is fundamental to tasks such as key identification and score transcription. These tasks in general depend on both a harmonic (tonal) analysis and a rhythmic (metrical) analysis.

Our focus in the present paper is on harmonic analysis. We use the three-dimensional tonal harmonic space first described by Euler (1739) and others, and developed, including the distance metric we use, by Longuet-Higgins (1962a,a) and Longuet-Higgins & Steedman (1971). This representation provides the basis for a theory of tonal harmonic progression—that is, a framework in which to analyse the relationship between the chords underlying a passage of music. We treat this analysis of the tonal relations between chords analogously to the logical semantics of a natural language sentence. By defining a representation of movements in the tonal space in a form similar to those used to represent natural language semantics, we are able to apply techniques from natural language processing directly to the problem of harmonic analysis.

We define a formal grammar of jazz chord sequences using a formalism based closely on one used for natural language processing. We then use statistically-based modeling techniques commonly applied to the task of parsing natural language sentences with such a grammar to map music, in the form of chord sequences, onto its underlying harmonic progressions in the tonal space.

We use supervised learning over a small corpus of chord sequences of jazz standards from the Real Book (Hal Leonard Corp., 2006) annotated by hand with harmonic analyses that we treat as a gold standard. We show that grammar-based musical parsing using simple statistical parsing models is more accurate than a baseline Markovian model.

2.2 The Relationship of Music and Language

The temptation to believe music and language to be closely related cognitive systems seems irresistible. For example, the early Sanskrit grammarians noticed strong parallels between their language and their music at the level of the sound-systems of phonology and prosody (Daniélou, 1968, cf. Lerdahl & Jackendoff, 1983:314-330). At times, this insight has led to the application of theoretical devices from language to music. For example, the "autosegmental-metrical" theory of spoken prosody draws explicitly on the musical distinction between rhythmic and metric levels of analysis (Liberman, 1975), while the "dynamic programming" algorithms that are used to rhythmically align spoken English phrases to their stress-timed metre work have been successfully applied to musical phrasing (Temperley, 2007).

It has been much less clear how to extend this apparently productive generalization to higher levels of structure and interpretation. While anglophone linguists often talk as if surface structural constituency and its recursive nature were properties shared by all languages, self-evident from examination of their stringsets, it is obvious that such claims, however reasonable they seem in application to English, can at best only be half-truths when applied to "free word-order" languages such as classical Latin, Turkish, and Warlpiri. In fact, the only sound basis for belief in recursive constituent structure in any language, including English, is that we all share some very strong (albeit incomplete) intuitions about the underlying *semantics*.[1]

Those intuitions leave us no reasonable option but to believe that the *meaning* of a sentence like *I told Warren that Dexter said he was ill*, regardless of the alignment of that meaning with the words of the language in which it is uttered, is that of *a proposition about another proposition* (which in turn is about yet another proposition), and that such semantic embedding can be of unlimited depth.[2]

Our intuitions about linguistic meaning may not always be sound, and they are certainly incomplete, but they give us something to go on. Unfortunately, any comparably strong intuitions about the underlying meaning of music seem

[1] It is often assumed that one can decide the automata-theoretic class of of a language on the basis of its stringset alone. However, this is not in practice possible (Savitch, 1989).

[2] If we want to claim in addition that English has a "surface" syntactic structure that exhibits quite a lot of the same constituency, we should be clear that we are making a language-specific claim about the *derivation* of such meaning representations, which will not hold for all languages, and may therefore be extremely misleading, even for English.

almost entirely lacking. For example, what is the "meaning" of the melodic passage in figure 2.1 (first used by Longuet-Higgins (1979))?

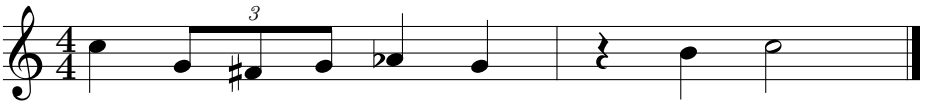


Figure 2.1: "Shave and a haircut, six bits"

It is clearly rhythmically and melodically well-formed and complete. It is, in its little way, quite satisfying as a piece of Western tonal music, in a sense that, say, the first six notes alone would not be. In that sense, we may be tempted to assign a rhythmic and harmonic structure to it, say along the lines suggested by Lerdahl & Jackendoff. However, such structures will only be interesting to the extent that they have an interpretation.

The distinction between interpretable structure and other kinds of uninterpreted structure, such as grouping and repetition, is very important for this discussion. A sonata undoubtedly has a structure, but one may doubt whether such structure is interpretable at the highest level, yielding a unified meaning for the whole. One's doubts arise for much the same reason one doubts that *War and Peace* is dominated by the root of a gigantic tree corresponding to the meaning of the entire novel. Impatience with the Schenkerian tendency to equate all levels of structure, together with the failure of Shenkerian analysts to provide convincing rules for automatically constructing (let alone interpreting) their structures, lies at the heart of criticisms of the approach like that of Narmour (1977), and have even led critics like Tymoczko (2011) to deny that musical structure is recursive at any level.

Of course, if we are talking about structures like sonata form, or quasi-phonological processes like voice-leading, Tymoczko is quite right. However, it would be wrong to conclude that there is *no* level of structure in music that is recursive, for the same reason that it would be wrong to infer from evidence for lack of recursion in the structure of War and Peace or vowel harmony that no aspect of language is recursive. What we should ask instead is whether there is any level of musical structure that is recursive. Moreover, as in the case of language, we should understand that the question fundamentally concerns musical semantics. We will be inclined to accept recursion in the syntax of music to the extent that we can identify recursion in its semantics.

The notion of musical meaning has very frequently been linked to the idea of the emotions (Cooke, 1959). However, most theories of the emotions regard them as secondary to other psychological events, whether autonomic, as in James (1884), or intensional—that is, *about some fact or proposition*, as proposed by Sartre (1947). The most empirically testable claims of this kind concerning music have focused on the satisfaction or frustration of melodic *expectations* of various kinds (Meyer, 1956; Cooper & Meyer, 1963; Narmour, 1977; Lerdahl & Jackendoff, 1983; Margulis, 2005; Huron, 2006).

In this connection, it may be helpful to reconsider figure 2.1. It seems intuitively obvious that this little piece consists of two parts, corresponding to the two bars, and that the first bar creates an expectation which the second bar satisfies. More specifically, the first bar moves from the tonic or key note C to

the fifth or dominant tonality of G, which creates an expectation of a cadential return to the tonic.

The second bar could satisfy this need in a number of ways, such as a whole-note C, or another G followed by C, but is particularly satisfying as written because it moves *via a diatonic semitone* from a B (which is still in the dominant tonality) to the expected tonic C, a simple example of voice-leading. To the extent that there is any wit in the piece at all, it resides in the further detail that this resolution is accomplished, not on the metrically prominent first beat of the bar, but on the weak third beat. This syncopation prolongs the suspense, increasing the pungency of its eventual fulfilment.

The above description can be verified by making the claimed tonal progressions explicit with some chords, namely a chord of C major for the first half bar, establishing the tonic, progressing to G major (with an added "dominant" seventh F – G^7) in the second half-bar, then (after the quarter note rest) a further chord of G^7 followed by C major in the second bar. In the light of this observation, we can claim that an important part of the meaning of the piece as a whole is a statement of the tonic, followed by a "cadential" progression from its dominant back to that tonic.

Where we see semantics, we must expect to find syntax to build it. We can meaningfully ask what musical language such simple sequences are drawn from, and what its grammar is, including whether it is recursive, and supports other phenomena characteristic of natural language, such as the long-range dependencies exhibited in relativization and coordination. We shall see that for this kind of musical sequence, which a jazz musician might identify as a "theme", and use as a basis for improvisation, there is a grammar consisting of a syntax of a linguistically familiar "near context-free" formal class, with a fully compositional semantics, complete with the lineaments of a model theory.

In this connection, it is also interesting to observe that the mnemonic phrase in the caption to figure 2.1, which is often used to refer to this tune, is structured linguistically as a "topic", stating the service concerned, and a "comment", concerning its price, rather than as a sentence as such, with a finite predication. We should therefore keep an open mind as to exactly what aspects of language grammar will be found mirrored in music, and what will be the best formalism for capturing them.

2.3 Musical Syntax

The syntax of tonal harmony and that of natural language can both be analysed using tree structures, and both have been claimed to feature formally unbounded embedding of structural elements (Keiler, 1981; Lerdahl & Jackendoff, 1983; Rohrmeier, 2011).

2.3.1 Cadences

The key component of harmonic structure is the *cadence* of a kind implicit in figure 2.1, built from tension-resolution patterns between chords. Large structures can be analyzed as *extended cadences*, made up of successive tension-resolution patterns chained together.

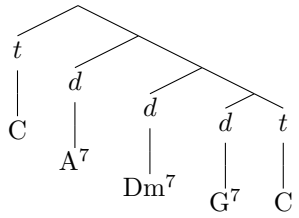


Figure 2.2: An extended authentic cadence, a typical example of tail recursion in music. The A^7 acts as a dominant resolving to the Dm^7, which in turn resolves by the same relation to G^7, which then resolves to the tonic C.

Cadences come in two varieties. The *authentic cadence* consists of a tension chord rooted a perfect fifth above its resolution. This type of tension chord is referred to as a *dominant* chord, and is the kind illustrated in figure 2.1. The *plagal cadence* consists of a tension chord rooted a perfect fourth above its resolution. This type of tension chord is referred to as a *subdominant* chord. In both cases, the resolution chord is classified as a *tonic* chord. This classification of a particlar occurrence of a chord identifies its *function* on that occasion of use, and partly establishes its place in the harmonic structure in relation to the surrounding chords. The same chord type, such as a G major triad, may function either as a dominant or subdominant tension chord, or as a tonic resolution, on different occasions of use in the same piece.

An extended cadence occurs when a tension chord resolves by the appropriate interval to a chord that is itself cadential, creating a further tension and subsequently resolving. Such a definition is recursive, and extended cadences can accordingly be indefinitely extended. This may be done with either type of cadence, but is most common with the authentic cadence. An example is shown in figure 2.2.

Such extended cadences are (rightward) "tail-recursive", and the recursion can therefore be "covered" or simulated by a finite state machine. Nevertheless, the semantics requires the embedding shown in figure 2.2.

A cadence may not reach its eventual resolution in a tonic chord immediately. An *unresolved* dominant (or subdominant) cadence, such as Dm7 G7, creating an expectation of tonic C, may be interrupted by a further cadence, A7 Dm7 G7, creating the same expectation, whereupon *both* cadential expectations/tensions will be resolved by the *same* tonic C, as in

C (Dm7 G7) (A7 Dm7 G7) C

We refer to this operation as *coordination* by virtue of its similarity to right-node raising coordination in sentences like *Keats bought and will eat beets* (analyzed below), in which *beets* satisfies the expectations of both *bought* and *eat*.

Coordinated cadences may themselves be embedded in coordinated cadence, as in this example from *Call Me Irresponsible*, with coordination of constituents marked by &:

$((G\sharp\circ^7\ Am^7\ Dm^7)\ \&\ ((A^7\ \&\ (E\phi^7\ A^{7\flat 9}))\ D^7))$

The full cadence, which includes still further levels of embedding, is shown in a tree in figure 2.3.

This process is again mirrored in natural language examples like *Keats ((may*

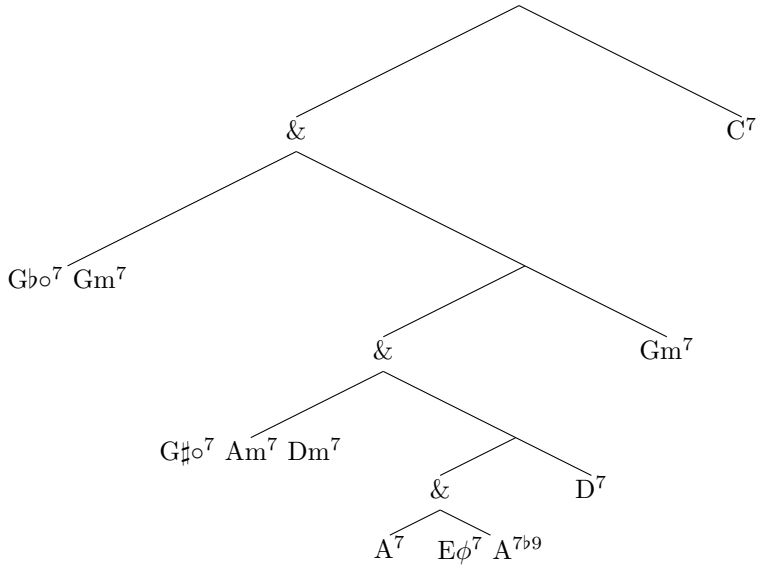


Figure 2.3: Tree representing the embedded structure of unfinished cadences in *Call Me Irresponsible*. The cadence shown here is in fact further embedded: the eventual resolution to the tonic F is not reached until after another cadence structure similar to this one.

or may not) cook) but (certainly eats) beets.

Chords that function as dominants are often partially, though never unambiguously, distinguished by the addition of notes other than those of the basic triad. In particular, the "dominant seventh", realized by the note two semitones below the octave, enhances the cadential function of a dominant chord and heightens the expectation of the corresponding tonic. However, this note may be omitted from a dominant chord, and conversely the same note (or rather, one indistinguishable from it on the equally tempered keyboard) may be used in chords that are not functioning as a dominant.

2.3.2 The Jazz Sublanguage

The typical size and complexity of the cadence structures discussed above varies with musical period and genre. Tonal jazz standards or themes are of particular interest for this form of analysis for several reasons.

First, they tend to feature large extended cadences, often with complex embedding. Second, they contain many well-known *contrafacts*, harmonic variations of a familiar piece, created using a well-established system of harmonic substitutions, embellishments and simplifications.

Finally, jazz standards are rarely transcribed as full scores, but are more analystically notated as a melody with accompanying chord sequence. Analysing the harmonic structures underlying chord sequences, rather than streams of notes, avoids some difficult practical issues such as voice leading and performance styles, but still permits discovery of the kind of higher-level structures we are concerned with.

Our study focusses on the analysis of harmonic structure in chord sequences

of jazz standards. This is not to say that the approach to analysis is not applicable beyond this domain or even that it depends on analysing chord sequences. The lexicon of the grammar outlined below, however, is somewhat specific to the genre.

In our ongoing work, we intend to extend the models described in this paper to perform the same analysis on note streams. Chapter 4 presents some ideas that we hope will form the basis for such a model.

2.4 A Model of Tonality

In analysing the roles of pitch in music, it is important to distinguish between *consonance*, the sweetness or harshness of the sound that results from playing two or more notes at the same time, and harmony, which is the dimension relevant to the phenomenon that we have already alluded to as tension (and the creation of expectation) and resolution (or its satisfaction). Both of these relations over pitches are determined by small whole-number ratios, and are often confounded. However, they arise in quite different ways.

2.4.1 Consonance

The modern understanding of consonance originates with Helmholtz (1862), who explained the phenomenon in terms of the coincidence and proximity of the secondary overtones and difference tones that arise when simultaneously-sounded notes excite real non-linear physical resonators, including the human ear itself, inducing harmonics or secondary tones. To the extent that an interval's most powerful secondary tones exactly coincide, it is perceived as consonant or sweet-sounding. To the extent that any of its secondaries are separated in frequency by a small enough difference to "beat" at a rate which Helmoltz puts at around $33Hz$, it is perceived as dissonant, or harsh.

Thus, for the diatonic semitone, with a frequency ratio of 16/15, only very high-frequency, low-energy overtones coincide, so it is weakly consonant, while the two fundamentals themselves produce beats, in the usual musical ranges, so it is also strongly dissonant. For the perfect fifth, on the other hand, with a frequency ratio of 3/2, all its most powerful secondaries coincide, and only very weak ones are close enough to beat. The fifth is therefore strongly consonant and only weakly dissonant.

This theory, has survived to the present day, and, with an important modification to incorporate Plomp and Levelt's 1965 "critical bandwidth" (which reflects the neurocomputational nature of the cochlea and auditory cortex), successfully explains the subjective experience of consonance and dissonance in chords, and the effects of chord inversion.

2.4.2 Harmony

The tonal harmonic system also derives from combinations of small integer pitch ratios. However, the harmonic relation is based solely on the first three primes ratios in the harmonic series: ratios of 2, 3 and 5 (commonly known as the octave, perfect fifth and major third). The tuning based on these intervals is known as *just intonation*.

E^-	B^-	$F\sharp^-$	$C\sharp$	$G\sharp$	$D\sharp$	$A\sharp$	$E\sharp^+$	$B\sharp^+$
C^-	G^-	D^-	A	E	B	$F\sharp$	$C\sharp^+$	$G\sharp^+$
$A\flat^-$	$E\flat^-$	$B\flat^-$	F	C	G	D	A^+	E^+
$F\flat^-$	$C\flat^-$	$G\flat^-$	$D\flat$	$A\flat$	$E\flat$	$B\flat$	F^+	C^+
$D\flat\flat^-$	$A\flat\flat^-$	$E\flat\flat^-$	$B\flat\flat$	$F\flat$	$C\flat$	$G\flat$	$D\flat^+$	$A\flat^+$

Figure 2.4: (Part of) The Space of Note-names (Longuet-Higgins 1962a,b)

Just Intonation

In just intonation, an interval can be represented as a frequency ratio defined as the product $2^x \cdot 3^y \cdot 5^z$, where x, y, z are positive or negative integers. It has often been observed since Euler (1739) that the harmonic relation can therefore be visualized as an infinitely extending discrete three-dimensional space with these three prime factors as generators. Since notes separated by octaves are essentially equivalent for tonal purposes, it is convenient to project the space onto the $3, 5$ plane.

We present this theory as it was formally developed by Longuet-Higgins (1962a,b) in figure 2.4.

Longuet-Higgins & Steedman (1971) observed that all diatonic scales are convex sets of positions, and defined a Manhattan taxi-ride distance metric over this space. According to this metric, it will be observed that the major and minor triads, such as CEG and CE♭G, when plotted in this space are two of the closest possible clusters of three notes. The triad with added major seventh is the single tightest cluster of four notes. The triads and the major seventh chord are stable unambiguous chords that raise no strong expectations and are of the kind that typically end a piece. Chords like the diminished chord and the dominant seventh are more spread out. We shall see that this difference between the two kinds of chord is vital to the induction of harmonic expectation, and its satisfaction.

The space of justly tonal intervals does not include ratios involving higher prime factors. For example, the closest musical interval to a ratio of $\frac{7}{8}$ relative to an origin C is B♭. However, none of the intervals named B♭ in the portion of the justly-intoned tonal space shown in figure 2.4 is as close in frequency to $\frac{7}{8}\cdot$C as they are to each other.

There is a great temptation in contemplating this spatial representation of the harmonic relation to start tinkering with it, to make it express other relations of pitch, such as consonance. For example, the major and minor triads are extremely consonant, but the other two closest clusters of three notes, such as CEB and CGB are dissonant, because they involve the interval CB. If we make the harmonic space into a triangular space, as Euler (1739), Riemann (1914), and the neo-Riemannians do (Cohn, 1997), then the major and minor triads become the sole closest clusters of three notes: CEB and CGB have a larger summed Mannhattan distance. Thus it seems as if we might be able

to capture both harmony *and* consonance in a single Eulerian *Tonnetz*, as in figure 2.5.

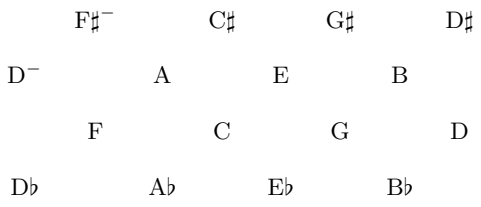


Figure 2.5: (Part of) The Neo-Riemannian *Tonnetz* (Cohn, 1997)

However, this temptation should be resisted. Such a representation greatly weakens the harmonic faithfullness of the metric implicit in figure 2.4. The *Tonnetz* falsely equates the harmonic character of the minor tone C,D$^-$ and the chromatic semitone E♭,E, which are intervals of size 3 in figure 2.4, with those of the harmonically closer major tone, C,D and diatonic semitone, D♯,E, both of size 2. In the *Tonnetz*, all four intervals have the same size, 2. This means that the *Tonnetz* fails to account for the stronger voice-leading effect of the harmonically closer major tone and semitone. (Tymoczko, 2011, Appendix C, points out the shortcomings of the *Tonnetz* as a theory of voice-leading.)

Equal Temperament

Over several centuries, it was gradually realised that the tonal harmonic space could be approximated, first by slightly mistuning the fifths to equate all the positions with the same label in figure 2.4, and then more by even further distorting the major thirds, to equate C with B♯, D♭♭. This is done by spacing the 12 tones of the diatonic octave evenly, so that all the semitones are (mis)tuned to the same ratio of $\sqrt[12]{2}$.

Since the eighteenth century most instruments have been tuned according to this system of *equal temperament*, which has the advantage that all keys and modes can be played on the same instrument without retuning. In terms of the tonal space, the result is a distortion of the pitches so that the infinite space is projected onto a finite toroidal space of just 12 points, looping in both dimensions. Each point is (potentially, infinitely) tonally ambiguous as to which point in the full justly-intoned space of figure 2.4 it denotes.

Equal temperament thus obscures the tonal relations underlying the tuning system, making their interpretation in terms of the original justly-intoned intervals ambiguous. The advantage of equal temperament, however, is that it allows the hearer to *resolve* this tonal ambiguity, and invert the projection onto the torus to recover the interpretation of the intervals in the full harmonic space. This is possible because the harmonic intervals that are sufficiently close in justly intoned frequency to be equated on the equally tempered torus *are sufficiently distant in the full space for the musical context to disambiguate them.* For example, if I have decided that my origin is justly-intoned G, then an equally tempered note that could in isolation be interpreted as any of C, C$^-$, C$^+$, B♯, D♭♭, etc. must be interpreted as C, because that is the only harmonic interpretation that is anywhere close to G.

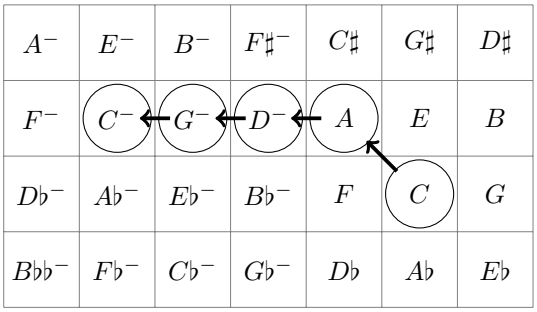


Figure 2.6: A tonal space path for the extended cadence: C A^7 D^7 Gm^7 C. Note that the path ends at a different C to the origin. The two are not distinguished by equal temperament.

It is important to realize that ambiguous equally-tempered music is unconsciously interpreted in terms of the full tonal space of harmonic distinctions, just as a (theoretically, infinitely ambiguous) two-dimensional photograph is interpreted as a three-dimensional scene. It is for this reason that equally-tempered B♭ is interpreted in tonal music as related to C by either a dominant seventh (ratio $\frac{8}{9}$) or a minor seventh (ratio $\frac{8}{9}$), but never by an interval related to the seventh harmonic (ratio $\frac{7}{8}$), as noted above. The equally-tempered minor/dominant seventh should therefore never be claimed to approximate a suboctave of the seventh harmonic, as is often alleged (Jeans, 1937; Bernstein, 1976; Tymoczko, 2006).[3]

2.4.3 Domain for Analysis

In our grammar for jazz chord sequences, we therefore take the tonal space as the semantic domain of harmonic analysis. The harmonic interetation of a piece is the path through the tonal space traced by the roots of the chords.

If we establish that there is a dominant-tonic relationship between two chords, we know that the underlying interval between the roots is a perfect fifth, a single step to the left in the space. Likewise, a subdominant-tonic relationship dictates a perfect fourth, a rightward step. Where no tension-resolution relationship exists, as between a tonic and the first chord of a cadence that follows it, we assume a movement to the most closely tonally related instance of the chord root.

Figure 2.6 shows an example of a tonal space path for an extended cadence. The perfect fifth relationship between the dominants and their resolutions is reflected in the path. The first step on the path is not a tension-resolution relationship, so proceeds to the closest instance of the A.

In this way, by identifying the syntactic structure of the harmony, that is the recursive structure of tension-resolution relationships between pairs of chords, we produce the path through the space that this dictates for the chord roots of

[3]This is not of course to deny that varieties of music *other* than the tonal might take the seventh harmonic as a primitive ratio, although it is doubtful that such a music could support equal temperament or even a very extensive form of harmony. The Bohlen-Pierce scale (Mathews & Pierce, 1989), which is based on prime ratios 3, 5 and 7, appears to prove the point.

the progression. We recover the underlying justly intoned pitches of the music which were realised as their projection onto equal temperament approximations.

2.5 Combinatory Categorial Grammar

2.5.1 CCG for English

In this section, we give an overview of the grammar formalism of Combinatory Categorial Grammar (CCG) in its conventional application to natural language, covering only aspects that are relevant to its application to music in the next section, where we describe its adaptation to express a syntax of tonal harmony. See Steedman (2000) to more details of CCG for language.

The derivation of the syntactic structure of a sentence using CCG is performed by assigning a complex syntactic category to each word in the input which determines what sorts of structure the word may appear in. The category is chosen from a large lexicon. Consecutive categories are combined using a small set of grammatical rules, constrained by the form of the categories, eventually producing a single category spanning the whole input.

A small set of atomic categories is used, including, for instance, S (sentence) and NP (noun phrase). Complex categories are built from atomic categories using the $/$ and $\backslash$ operators. A complex category X/Y denotes a *function* category that can combine with an *argument* category Y to its right to produce a result of category X. Likewise $X\backslash Y$ indicates that a Y is expected to the left.[4]

Categories can combine by grammatical rules of *function application*, defined as:

a. $X/Y \quad Y \quad \Rightarrow \quad X \quad (>)$
b. $Y \quad X\backslash Y \quad \Rightarrow \quad X \quad (<)$

Figure 2.7 shows the use of the function application rule in a simple derivation.

$$\dfrac{\dfrac{\text{Keats}}{NP} \quad \dfrac{\dfrac{\text{eats}}{(S\backslash NP)/NP} \quad \dfrac{\text{beets}}{NP}}{S\backslash NP}\,{>}}{S}\,{<}$$

Figure 2.7: A simple syntactic derivation using the forward and backward function application rules.

In order to produce a compositional denotational semantics for the full sentence from the syntactic derivation, each lexical item also has a logical form and each rule defines how the logical forms of its arguments are combined.

a. $X/Y : f \quad Y : x \quad \Rightarrow \quad X : f(x) \quad (>)$
b. $Y : x \quad X\backslash Y : f \quad \Rightarrow \quad X : f(x) \quad (<)$

Figure 2.8 shows an example of a derivation with a logical form associated with each category.

[4]This is the "result leftmost" notation. There is an alternative "result on top" notation, due to Lambek.

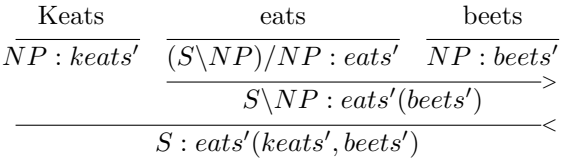


Figure 2.8: A CCG derivation including logical forms.

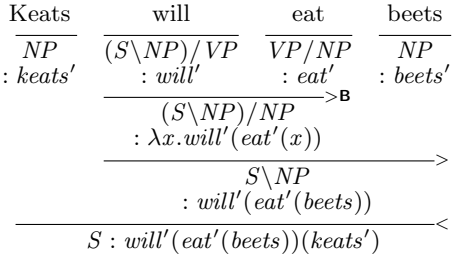


Figure 2.9: A syntactic derivation making use of the composition rule to combine *will* with *eats* before reaching *beet*.

Several other rules allow grammars to capture linguistic phenomena such as coordination and relativization. The only one relevant to the present discussion is *composition*.

Composition permits complex categories to be combined before their argument is available. The result may then be applied (using function application) to the argument when it is eventually encountered. The final outcome is the same as if only function application had been used, but composition allows this outcome to be produced by a different order of combinations. This is important for, among other things, incremental analysis of a sentence.

Forward Composition (> **B**):

$$X/Y \ : \ f \quad Y/Z \ : \ g \quad \Rightarrow_{\mathbf{B}} \quad X/Z \ : \ \lambda x.f(g(x))$$

Figure 2.9 demonstrates the use of the composition rule.

It should be noted that, although in this particular derivation the analysis of the tensed verb phrase is left-branching, the logical form that it builds is right branching and identical to that in the alternative application-only derivation, as its semantics requires. (The latter is suggested as an exercise.)

The full range of reasons for treating natural language grammar in this way need not detain us here, but one is to do with the fact that constructions like coordination involving long-range semantic dependencies treat incomplete fragments like *will eat* as typable *constituents* that can be combined with others of the same type in derivations like figure 2.10 (the semantics is omitted, but can be inferred from figure 2.9).

Such examples call for a generalized notion of natural language surface structure with a monotonically compositional semantics. The present paper shows that musical analysis also involves long-range dependencies, and seems to call for the same approach.

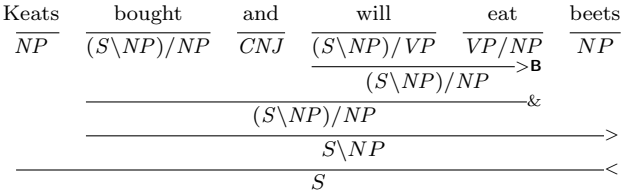


Figure 2.10: A syntactic derivation using the coordination rule to combine *bought* and *will eat* into a single constituent that can combine with *beats*.

2.5.2 CCG for Harmony

For parsing the syntax of harmony, we use a formalism similar to the standard CCG for English described above. Instead of the linguistic syntactic categories, such as NP and S, we use harmonic syntactic categories that define cadential expectation. We use some of the standard combinatory rules from CCG, namely those mentioned in the introduction above, and some new rules specific to harmonic syntax.

An atomic category carries information about the harmony at the start and end of the passage it spans. This is the only harmonic information relevant to constraining how it can combine with adjacent categories. Each end has a harmonic root, in the form of an equally tempered note, and a chord function, one of T (tonic), D (dominant) and S (subdominant).

A passage beginning on a tonic on F and modulating to C receives the category $F^T\text{–}C^T$. The two-step cadence Dm^7 G^7 C would receive the category $D^D\text{–}C^T$. Each such assignment represents an interpretation of the harmonic structure of the sequence. Typically a sequence of chords specified by equally tempered notes will support many such interpretations, varying in plausibility. In the case of the latter sequence, we have decided that the C is a tonic chord, the Dm^7 a second-level dominant and the G^7 a first-level dominant.

A forward-facing slash category X/Y gives the starting point Y expected for the category to its right (its argument part, written after the slash) and the starting point X that will be used for the result of applying it to such an argument. Such a category interprets a dominant chord, like those in the example in figure 2.11. For brevity, where the start and end parts of an atomic category are the same, we write just one: a category $C^T\text{–}C^T$ is abbreviated to C^T. Such a category interprets a tonic chord.

Extended cadences, such as the one in figure 2.11, are handled by allowing the dominant category's argument to be itself a dominant. Here we are able to use composition, just as in the linguistic examples in section 2.5.1. The dominant categories can be combined into a single category, still a forward slash category, which is then combined with the resolution. It should be noticed that this particular derivation of the extended cadence is now left-branching, like the linguistic example 2.9.

Backward slash categories are precisely the reverse of forward slash categories. They specify the end point required of the argument (after the slash) and the end point that the result will have.

A combinatory rule resembling a generalization of the one used for natural language coordination in figure 2.10 allows interpretation of interrupted, or *co-*

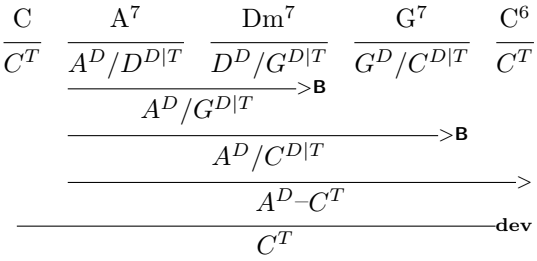


Figure 2.11: Derivation tree of an extended cadence. The three dominant chords are all interpreted using the same category. They are combined by composition first, then with their resolution by application. Finally, the initial tonic is combined with the resolved cadence using the development rule (see section 2.6.2).

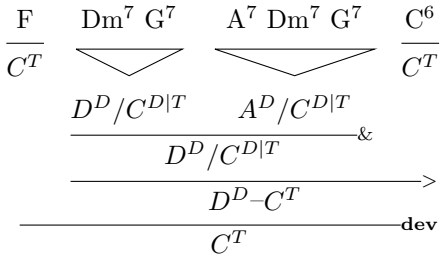


Figure 2.12: Derivation tree of a coordinated extended cadence. Details of the derivations of the cadence constituents are omitted for brevity. The constituents can be combined using coordination (marked &) because they are looking for the same resolution.

ordinated, cadences. A chain of dominant seventh chords, without their final resolution, can be treated as a constituent, thanks to composition. The coordination rule combines two such constituents which expect the same resolution into a single slash category, which also expects this resolution. An example is shown in figure 2.12.

Chord substitution[5] is handled by the lexicon. For example, a seventh chord may be interpreted as a dominant chord rooted on the augmented fourth of the root as played. This is the *tritone substitution*, common in jazz. Other similar substitutions are handled similarly by adding a new line to the lexical schema in figure 2.13.

An example of this is found in this cadence from *Can't Help Lovin' Dat Man* (in the key of E♭), where the B^7 replaces a Fm^7 by the tritone substitution:

Gm^7 Cm^7 B^7 B♭aug^7 B♭^{7} E♭^{6}

[5] *Substitution* refers to the musical term for replacement of one chord by another functionally equivalent chord and is not related to the substitution rule of standard CCG.

2.6 A Grammar for Jazz

2.6.1 The Lexicon

We are now able to define the jazz chord lexicon in full, as in figure 2.13. Each entry has a mnemonic label to serve as an identifier, a surface chord type, a syntactic type and a logical form. The surface chord type generalizes over chord roots X. During parsing, a lexical schema may be used to assign a category to a chord, provided the chord falls into the general class of chords represented to the left of the :=. The roots within the category itself (right of the :=) are expressed here, using roman numerals, relative to the played root of the chord to which the category is assigned.

Thus, the mnemonic *Dom* identifies a rule that says a surface chord C^7 can be interpreted with the syntactic type $C^D/F^{D|T}$ and a logical form denoting a leftward step in the space.

All surface chords are in equal temperament. It is therefore meaningless to distinguish between enharmonic equivalents, like G♯ and A♭. For consistency, we arbitrarily choose to use flats throughout the lexicon.

The mnemonic label *Ton* is used to identify a simple tonic chord function. The corresponding syntactic category takes on the same root that the chord had. The logical form represents a point in the tonal space which is constrained to be one of those points that are mapped by equal temperament to the root of the surface chord. At this stage, we cannot say which of this infinite set of points this point is – this will be determined by the other constraints imposed by the full logical form of the sequence. Like the syntactic types, this coordinate of this point implicitly generalizes over the possible roots that the surface chord may have. For example, if the surface chord has root C, the logical form will become $\langle 0, 0\rangle$, whilst if the root is B the logical form is $\langle 1, 1\rangle$ (see figure 2.4).

The mnemonic label *Dom* identifies a simple dominant chord function. Applied, for example, to a surface chord G^7, it assigns syntactic type $G^D/C^{D|T}$. Its semantics is a single step leftwards in the tonal space, landing on its resolution. As in the case of natural language semantics, we use the lambda calculus to write a functional semantics. When one of these categories is combined with its resolution, the predicate will be applied to the resolution's logical form. *left-onto* and *rightonto* predicates represent a point constrained to be one step to the right and left (respectively) of their argument.

Figure 2.14 show this in action, via the action of the combinatory rules defined in the next section. It should be noticed that the cadential logical form is right-branching like figure 2.2, despite the fact that the syntactic derivation that builds it is left-branching. We give a fuller explanation of the semantics in chapter 3.

The mnemomic *Dom-tritone* identifies the tritone substitution of a dominant function chord. Such a chord behaves just like a dominant chord, but has a root six semitones away from the root that the chord appears to have. Indeed, the syntactic type is identical to that that would have been assigned as a simple dominant interpretation of the substituted chord (that rooted on the tritone). In other words, this entry allows us to interpret a chord $D♭^7$ exactly as if it had been a G^7 chord.

Mnemonic functional label	**Surface chord type**		**Syntactic type**		**Logical form**
Ton.	X(m)	:=	I^T	:	$[\langle 0,0\rangle]$
Ton-III.	Xm	:=	$\flat VI^T$	:	$[\langle 0,2\rangle]$
Ton-bVI.	X	:=	III^T	:	$[\langle 0,1\rangle]$
Dom.	X(m)^7	:=	$I^D/IV^{D\|T}$	:	$\lambda x.leftonto(x)$
Dom-backdoor.	X(m)^7	:=	$VI^D/II^{D\|T}$	:	$\lambda x.leftonto(x)$
Dom-tritone.	X(m)^7	:=	$\flat V^D/VII^{D\|T}$	:	$\lambda x.leftonto(x)$
Dom-bartok.	X(m)^7	:=	$\flat III^D/\flat VI^{D\|T}$	:	$\lambda x.leftonto(x)$
Subdom.	X(m)	:=	$I^S/V^{S\|T}$	:	$\lambda x.rightonto(x)$
Subdom-bIII.	X	:=	$VI^S/III^{S\|T}$	:	$\lambda x.rightonto(x)$
Dim-bVII.	X$\circ$	:=	$IV^D/\flat VII^{D\|T}$	:	$\lambda x.leftonto(x)$
Dim-V.	X$\circ$	:=	$II^D/V^{D\|T}$	:	$\lambda x.leftonto(x)$
Dim-III.	X$\circ$	:=	$VII^D/III^{D\|T}$	:	$\lambda x.leftonto(x)$
Dim-bII.	X$\circ$	:=	$\flat VI^D/\flat II^{D\|T}$	:	$\lambda x.leftonto(x)$
Pass-I.	X$\circ$	:=	I^T/I^T	:	$\lambda x.x$
	X$\circ$	:=	I^D/I^D	:	$\lambda x.x$
Pass-VI.	X$\circ$	:=	VI^T/VI^T	:	$\lambda x.x$
	X$\circ$	:=	VI^D/VI^D	:	$\lambda x.x$
Pass-bV.	X$\circ$	:=	$\flat V^T/\flat V^T$	:	$\lambda x.x$
	X$\circ$	:=	$\flat V^D/\flat V^D$	:	$\lambda x.x$
Pass-bIII.	X$\circ$	:=	$\flat III^T/\flat III^T$	:	$\lambda x.x$
	X$\circ$	:=	$\flat III^D/\flat III^D$	:	$\lambda x.x$
Aug-bII.	X^7	:=	$\flat VI^D/\flat II^{D\|T}$	:	$\lambda x.leftonto(x)$
Aug-VI.	X^7	:=	$III^D/VI^{D\|T}$	:	$\lambda x.leftonto(x)$
Colour-IVf.	X(m)	:=	V^T/V^T	:	$\lambda x.x$
Colour-IVb.	X(m)	:=	$V^T\backslash V^T$	:	$\lambda x.x$
Colour-IIf.	X(m)	:=	$\flat VII^T/\flat VII^T$	:	$\lambda x.x$
Colour-IIb.	X(m)	:=	$\flat VII^T\backslash\flat VII^T$	:	$\lambda x.x$

Figure 2.13: The lexical schema for the jazz grammar

$$\dfrac{\dfrac{\dfrac{\mathrm{Dm}^7}{D^D/G^{D|T} : \lambda x.leftonto(x)} \quad \dfrac{\mathrm{G}^7}{G^D/C^{D|T} : \lambda x.leftonto(x)}}{G^D/C^{D|T} : \lambda x.leftonto(leftonto(x))}\,{>}\mathbf{B} \quad \dfrac{\mathrm{C}}{C^T : [\langle -1,0\rangle]}}{D^D\text{–}C^T : [leftonto(leftonto(\langle -1,0\rangle))]}\,{>}$$

Figure 2.14

2.6.2 Combinatory Rules

Most of the work of the grammar is done in the lexicon. Only four rules are used to build derivations.

Application and **composition** are merely adaptations of their conventional forms to the musical formalism.

Application:

$$X/Y \ : \ f \qquad Y–Z \ : \ x \ \Rightarrow_{>} \ X–Z \ : \ f(x)$$

$$X–Y \ : \ x \qquad Z\backslash Y \ : \ f \ \Rightarrow_{<} \ X–Z \ : \ f(x)$$

Composition:

$$X/Y \ : \ f \qquad Y/Z \ : \ g \ \Rightarrow_{>\mathtt{B}} \ X/Z \ : \ \lambda x.f(g(x))$$

$$X\backslash Y \ : \ g \qquad Z\backslash X \ : \ f \ \Rightarrow_{<\mathtt{B}} \ Z\backslash Y \ : \ \lambda x.f(g(x))$$

The **coordination** rule combines unresolved cadences to behave as a single unresolved cadence. The two cadences are required to be of the same type – either authentic (dominant function) or plagal (subdominant function). The result's logical form is a function that will be applied to the resolution, as in a normal cadence, and will ensure that they both cadences resolve to the same point. See chapter 3 for a formal definition.

Coordination:

$$X^f/Y \qquad Z^f/Y \ \Rightarrow_{\&} \ X^f/Y$$

where $f \in \{D, S\}$

The **development** rule joins together fully resolved passages, building an interpretation of a whole piece out of its consituent cadences. Its semantics (also found in chapter 3) is the concatenation of the two constituents, implicitly constraining the path to make the smallest possible jump between the end of the first contituent and the start of the second.

Development:

$$V–W \qquad X–Y \ \Rightarrow_{dev} \ V–Y$$

This may seem an excessively permissive rule. It permits any two consecutive passages interpreted individually as harmonically stable to be conjoined, no matter what keys they are in. Such passages include individual chords. In the extreme case, we could interpret as whole chord sequence as modulating to a new key with every new chord. However, modulation can occur freely in music at any time and in some cases even such an extreme interpretation in which each chord establishes a new key may be valid. We therefore do not use the grammar to put any restrictions on what sorts of modulation to permit. It is certainly true that some modulations are more common than others, and it is such preferences as this that are well captured by the sort of statistical parsing models we discuss below.

An example derivation using all four rules is shown in figure 2.15. Further explanation of the semantics is given in chapter 3.

2.7 Statistical Models

Just as with natural language parsing, the lexical ambiguity of interpretation of chord sequences prohibits exhaustive parsing to deliver every syntactically well-formed interpretation of a sequence. Moreover, even if exhaustive parsing were practicable, we would need a way to distinguish the most plausible interpretations among a huge number of possible interpretations.

It is usual in parsing natural language to use statistical models based on a corpus of hand-annotated sentences to rank possible interpretations. Such techniques can be used to limit search during parsing and speed up parsing by eliminating apparently improbable interpretations early in the process. Bod (2002); Honingh & Bod (2005) and Temperley (2007) have shown statistical techniques used for language parsing can be applied to chord sequence parsing and other tasks for folksong domains. This paper shows that such methods can be extended to the present class of musical grammars.

2.7.1 Jazz Corpus

To train our statistical models, we have constructed a small corpus of jazz chord sequences. The sequences are taken from Hal Leonard Corp. (2006). We excluded certain sequences that could not be analysed using the grammar described above. In some cases, this is due to limitations of the lexicon (e.g. rare substitutions not covered by figure 2.13), in others because the song is outside the intended domain of the grammar, such as Thelonious Monk's *Epistrophy*.

Every chord is annotated with a choice of category from the lexicon of the jazz grammar. Since CCG is a lexicalized grammar formalism and we only use a small set of combinatory rules, this assignment of categories to chords contains most of the information necessary to define a unique gold-standard parse. With the addition of annotations of the points where coordination occurs, the corpus implicitly contains a unique tonal space analysis of every sequence. A simple deterministic parsing procedure using the annotations produces the analysis for any annotated sequence.

The corpus consists of 76 annotated sequences, totalling roughly 3000 chords. It contains no heldout test set: all models are tested using cross-validation (see section 2.8.2).

2.7.2 Adaptive Supertagging

Supertagging is a technique, related to part of speech (POS) tagging, used for parsing with lexicalized grammars like CCG (Srinivas & Joshi, 1994). Probabilistic sequence models, using only statistics about short-distance dependencies, are employed to choose CCG categories from the lexicon for each word. Ideally the choice of category, representing for us most of the interpretation of a chord, would depend on analysis of more distant parts of the sequence, that is on long-distance dependencies. In practice, short distance statistics can usually quite reliably rule out the most improbable interpretations.

A bad choice of categories could make it impossible to parse the sequence. The *adaptive supertagging* algorithm (Clark & Curran, 2007) allows categories considered less probable by the supertagger to be used if necessary. First, the supertagger assigns a small set of most probable categories to each word and

the parser attempts to find a full parse with these categories. If it fails, the supertagger supplies some more, slightly less probable categories and the parser tries again. This is repeated until the parser succeeds or we give up (for example, after a set number of iterations). If multiple full parses are found, the single most probable one is chosen.

In practice, even using the small set of categories chosen by the supertagger, the number of possible derivation trees is prohibitively large for the parser to explore them all. The parser applies a *beam* during parsing – that is, it removes (or *prunes*) all but the most probable interpretations at each intermediate node of the derivation tree. It has no model of the probability of a derivation tree. It applies the beam on the basis of the confidence that the supertagger gave the categories at the leaves of the tree.

Many types of probabilistic sequence model can be used as a supertagging model. We use a hidden Markov model (HMM) in which states represent categories. The state emissions we model are not the chords themselves, but a combination of the chord type and the interval between this and the previous chord's roots. This has the effect of making the model generalize over absolute pitch.

The model is trained using maximum likelihood estimation on the annotated categories from the corpus described above.

The limited size of the corpus means that it does not contain enough training data to train models much more complex than this. Some initial experiments with higher-order Markov models (n-gram models) suggest that they do not perform any better than the HMM we use here when trained on this small corpus.

We refer to this model, consisting of the HMM supertagger used in adaptive supertagging with a parser using only a naive pruning technique, as ST+GRAMMAR.

2.7.3 Baseline Model

In an attempt to quantify the contribution made by restricting interpretations to those that are syntactically well-formed under the jazz grammar, we have constructed a model which produces tonal space interpretations without using the grammar. We use an HMM very similar to that used as a supertagger in ST+GRAMMAR, which directly assigns a tonal space point to each chord, instead of assigning categories to chords and parsing to derive a tonal space path.

The representation of the chord sequence is identical to the supertagger's. We begin by defining a naive procedure for constructing a simple tonal space interpretation for a chord sequence. For each chord, we choose from the infinite set of points mapped by equal temperament to the chord's root the point that is closest to the previous point on the path. The states of the model are constructed to represent deviations from this initial path.

There are two reasons why such deviations from the naive path occur in valid analyses. First, it may be because the correct disambiguation of the equal temperament note is not the point closest to the previous, as happens at points of coordination, where the resolutions of two cadences are constrained to be the same. Second, it may be because of substitutions (like the tritone substitution described in section 2.5.2), where the surface chord's root is not the true root of the chord in the analysis.

The state labels consist of two coordinates. One denotes the deviation from the naively-chosen point due to substitution and the other the disambiguation of the equal-temperament note. An unsubstituted chord has a substitution coordinate of $(0, 0)$, whilst a tritone subsituted chord has $(2, 1)$. Most of the time, the disambiguation coordinate is $(0, 0$, representing the closest possible point to the previous point on the path; deviations of more than one unit in either dimension are rare. The HMM only includes those states that it observes in the training data.

The model is trained in the same way as St+Grammar, only this time the training data is chord sequences paired with their annotated tonal space paths. We refer to this model as HmmPath.

Apart from the construction of the state labels, the new model is identical to the supertagger's HMM. Unlike the supertagger, this model is used to evaluate the most probable tonal space path directly from the observed sequence, without having to be filtered by the parser for grammaticality. St+Grammar will completely fail to assign a path in cases where a full parse cannot be found using any of the supertagger's category predictions. HmmPath will assign some path to any sequence, since it is not limited to returning grammatical interpretations.

2.7.4 Adaptive Supertagging with Backoff

St+Grammar applied to some sequences produces no interpretation, when the parser fails to find a full parse given the probable categories suggested by the supertagger. This means that however high quality the returned paths are, the overall f-score (see section 2.8.1) is inevitably pulled down by the failure to interpret the chords of the omitted sequences.

A third model combines the two in an aggressive form of *backoff*. First, St+Grammar is applied to a chord sequence. If this produces a result, that is used. If it fails, HmmPath is used instead. We refer to this combined model as St+Grammar+HmmPath.

Since the HMMs of the supertagger and HmmPath are similarly constructed, backoff to the HmmPath can be thought of roughly as removing the filter of grammaticality from the interpretation. We assume that any interpretation found by St+Grammar will be more accurate than the result of HmmPath and so never back off if St+Grammar can find an interpretation, regardless of how probable the model considers it.

2.7.5 Parsing Models

Hockenmaier & Steedman (2002) adapted the generative probabilistic parsing models of probabilistic context-free grammars (PCFG) to CCG. Using a corpus of parsed sentences, probabilities are estimated for expansions at internal nodes in the derivation tree. These probabilities are used to estimate a probability for every subtree produced during the derivation.

St+PCFG uses the supertagger with the adaptive supertagging algorithm as described above. During parsing, a probability is assigned to every internal node in the tree using the parsing model. As before, a beam is applied to internal nodes, keeping only the most probable derivations, now using the parsing model's probabilities.

Model	*P (%)*	*R (%)*	*F (%)*	**Cov. (%)**
St+Grammar	**89.9**	61.9	73.3	75
HmmPath	74.6	82.1	78.2	100
St+Grammar+HmmPath	81.7	88.0	84.7	100
St+PCFG	87.4	85.7	86.5	95
St+PCFG+HmmPath	85.3	**91.0**	**88.1**	100

Table 2.1: Evaluation of each model's prediction of tonal space paths using 10-fold cross-validation on the jazz corpus. For each model, we report *precision* (P), *recall* (R), *F-score* (F) and *coverage* (Cov.).

St+PCFG can still fail to produce a full parse, either because the parse runs out of time or because the correct categories are not suggested by the supertagger or fall outside the beam of the parsing model. As before, we can use the HmmPath model in the cases where St+PCFG finds no interpretation. We call this backoff model St+PCFG+HmmPath.

2.8 Experiments

2.8.1 Evaluation

We evaluate all models on the basis of the tonal space path to which they assign highest probability. Paths are first transformed from a list of tonal space coordinates to a list of vectors between adjacent points. This means that a path which makes an incorrect jump (for example, to an enharmonic equivalent of the correct point) is only penalised for that mistake and not for all subsequent points. Each point also has an associated chord function, which is included in the evaluation.

We align this path optimally with the gold-standard tonal space path from the annotated corpus (pre-processed in the same way). We report precision, recall and f-score of the aligned paths. Precision is defined as the proportion of points returned by the model that correctly align with the gold standard. Recall is the proportion of points in the gold standard that are correctly retreived by the model. F-score is the harmonic mean of these two measures.

$$P = \frac{Aligned}{Aligned + Inserted}$$
$$R = \frac{Aligned}{Aligned + Deleted}$$
$$F = 2 \times \frac{PR}{P + R}$$

Since the tonal space path retreived consists of two pieces of information, the coordinate (now the step vector) and a chord function, we allow the alignment to assign a score of 0.5 an alignment of only one of these. Without this modification, a model that was, for example, very good at recognising substitutions, but poor at identifying which chords we tonics would score very badly on precision and recall, since no alignment would be counted where only the coordinate was correct.

2.8.2 Model Comparison

All models were trained on the jazz corpus described above, containing 76 fully annotated sequences. Since we cannot afford to hold out a test set, we used 10-fold cross-validation. Each experiment is run 10 times, with $\frac{9}{10}$ of the data used to train the model and the remaining $\frac{1}{10}$ used to evaluate the trained model. This means that all data is used for evaluation, but no model is tested on data that it was trained on. We report the results combined from all partitions.

The evaluation of the tonal space path is performed in every case only on the path returned by the model with highest confidence.

2.9 Results

The results of the three experiments are reported in table 2.1.

St+Grammar produces high-precision results. This is because it can only produce results that are permitted by the grammar and fails when it can find no such result. The model's recall is low due to the low coverage of the test sequences. It only finds a result for 75% of them.

HmmPath has a lower precision, since the paths are produced by regularities in the short-distance relationships between chords, but do not need to be globally coherent. The recall is higher than St+Grammar, because it produces at least some result for every sequence, but precision is low.

St+Grammar+HmmPath has a lower precision than St+Grammar because it now backs off to HmmPath. The recall is very much higher because the poor quality paths that come from HmmPath backoff are very much better than no path at all.

St+Grammar+HmmPath has a higher f-score than the two previous models. Firstly, this shows that HmmPath is a reasonable model to back off to when no grammatical result can be found. Secondly, it shows that the use of a grammar to constrain the paths predicted by an HMM model substantially improves over the purely short-distance information captured by the model.

St+PCFG gives a further improvement in f-score and has better coverage than St+Grammar (now 95%), probably largely because it is less likely to remove useful partial parses when applying the beam. St+PCFG+HmmPath produces the best f-score by supplying a low-precision interpretation where St+PCFG would fail to give any interpretation.

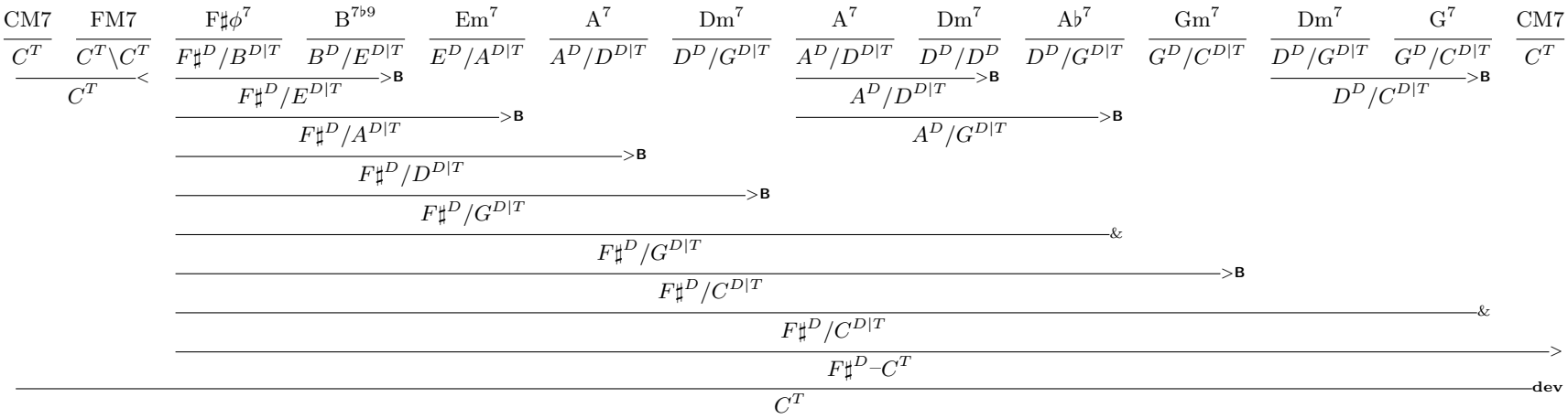


Figure 2.15: Syntactic derivation of a long extended cadence from *Alice in Wonderland* using the coordination and development rules. The logical forms are omitted. The same example is shown with its semantics in section 3.7.

2.10 Conclusion

The parser described above uses a formal grammar of a kind that is widely used for natural language processing, and statistically-based modeling techniques of a kind standardly used in wide-coverage natural language parsers, to map music onto underlying harmonic progressions in the harmonic space.

The corpus is small, but experience with CCG parsing for NLP shows that these techniques will scale to larger datasets (Clark & Curran, 2007; Auli & Lopez, 2011b).

The parsing model is built using supervised learning over a small corpus of jazz chord sequences hand-annotated with harmonic analyses. The fact that a grammar-based musical parser using a simple statistical parsing model is more accurate than a baseline Markovian model may be taken as further evidence suggesting that music and language have a common origin in a uniquely human system of interpersonal communication.

The parsing models we have described use a simple model based only on statistics over a short window of context as backoff. However, in most cases where the parser fails to find a full interpretation of a chord sequence, it does successfully identify large cadences, but cannot find an interpretation of certain difficult chords. We could construct a more cohent full analysis by identifying high-confidence partial analyses and backing off to a less constrained model only for those passages that proved difficult for the grammatical model. This appears to be a better emulation of what a human listener does on encountering a confusing passage of music, picking up the thread as soon as an easily identifiable tonal centre or cadence is heard.

The current paper describes models that analyze sequences of chords given in textual form. A certain amount of analysis has already been done in producing these chord symbols: the music has been divided into time segments during which the harmony remains constant; the most prominent notes have been selected; and the possible chord roots have been narrowed down somewhat. A model could be constructed that incorporates these tasks into the analysis process, accepting note-level input (in MIDI form, for example) and suggesting possible interpretations in the way the supertagger component of our parsing model does. Models exist to annotate MIDI data with chord labels. Raphael & Stoddard (2004) present a model that provides not only chord names, but a shallow form of harmonic analysis. We plan to experiment with using a model of this sort as the basis for a supertagger for MIDI data.

Chapter 3

A Tonal Space Semantics for Harmonic Analysis

3.1 Introduction

In chapter 2 we introduced a formalism for grammars of tonal harmony. The formalism, a modification of the Combinatory Categorial Grammar formalism for natural language, acts as a machanism to map a surface – chord or notes, in our case – onto a semantic interpretation – a tonal harmonic analysis. Each syntactic category is coupled with a logical form and, as syntactic categories are combined during parsing, a logical form representing the full harmonic analysis is built up.

In chapter 2 we noted that a logical form is constructed to represent a harmonic analysis in terms of movements about Longuet-Higgins' tonal space, but omitted the details of the representation we use. This chapter sets out the details of a representation suitable for our tonal semantics.

3.2 Tonic Semantics

The semantics of a tonic is a point in the tonal space. It is underspecified – it only specifies a point within an *enharmonic block* (see figure 3.1). It is therefore a coordinate between $\langle 0, 0 \rangle$ and $\langle 3, 2 \rangle$ and each coordinate denotes different infinite set of positions in the space.

A single tonic chord receives as its logical form a single-element list containing such a coordinate. A logical form of this sort is associated with atomic lexical categories, such as both the occurences of C^T in figure 2.11.

3.3 Cadence Semantics

The semantics of a cadence step is a predicate representing a movement in the tonic space. An extended cadence is interpreted as the recursive application of each movement to its resolution.

Authentic cadences – left steps – use the *leftonto* predicate and plagal cadences – right steps – the *rightonto* predicate. For example, a single dominant

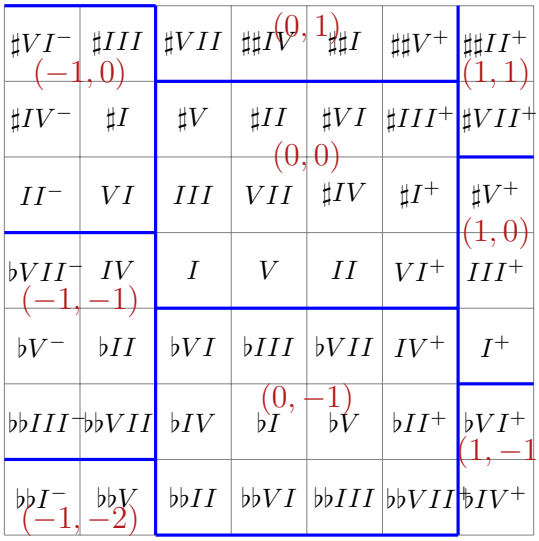


Figure 3.1: Enharmonic blocks at the centre of the space. Each position within these 4x3 blocks is equated by equal temperament with the same position within every other block.

chord resolving to a tonic $\langle 0,0\rangle$ would receive the logical form $leftonto(\langle 0,0\rangle)$, whilst a secondary dominant, resolving to a dominant, resolving to the tonic would receive $leftonto(leftonto(\langle 0,0\rangle))$.

We define *leftonto* (and likewise *rightonto*) as being subject to a reduction when applied to a list, as in the case of a tonic resolution, as follows:

$$leftonto([X_0, X_1, ...]) \Rightarrow [leftonto(X_0), X_1, ...]$$

Example (1) shows an example of a two-step cadence – the familiar IIm^7 V^7 I. The derivation shows the combination of the semantics of each chord into the semantics for the sequence.

Throughout this chapter, derivations like this are written with the syntactic part of each sign (syntactic type/logical form pair) omitted. Naturally, these are all derivations that would be permitted by the syntactic types associated with these logical forms under the combinators described in chapter 2.

(1)
$$\dfrac{\dfrac{\mathrm{IIm}^7}{\lambda x.leftonto(x)} \quad \dfrac{\dfrac{\mathrm{V}^7}{\lambda x.leftonto(x)} \quad \dfrac{\mathrm{I}}{[\langle 0,0\rangle]}}{[lefonto(\langle 0,0\rangle)]}>}{[leftonto(leftonto(\langle 0,0\rangle))]}>$$

The recursive application of multiple cadence steps can be combined ahead of time, before their application to their resolution, using the composition operator, associated with the composition combinator.

$$f \circ g \equiv \lambda x.f(g(x))$$

Example (2) shows the same interpretation as that in example (1) produced by a derivation that uses the composition combinator.

(2)
$$\dfrac{\dfrac{\dfrac{\text{IIm}^7}{\lambda x.leftonto(x)} \quad \dfrac{\text{V}^7}{\lambda x.leftonto(x)}}{\lambda x.leftonto(leftonto(x))}\,{>}\mathbf{B} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{[leftonto(leftonto(\langle 0,0\rangle))]}\,{>}$$

3.4 Colouration Semantics

The lexicon includes some categories for interpreting colouration chords, which contribute nothing much to the functional structure of the harmony, but spice up the realisation. Accordingly, these are given an empty semantics (that is, the identity function), which simply ignores them.

A typical example of this is the sequence *I IV I*, often played during long passages of a *I* chord. This is really a form of plagal cadence and a fine grained analysis might treat it as such. However, for most analysis purposes we wish to ignore this very brief excursion from the tonic. An example derivation using this empty semantics is shown in example (3).

(3)
$$\dfrac{\dfrac{\text{I}}{\lambda x.x} \quad \dfrac{\dfrac{\text{IV}}{\lambda x.x} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{[\langle 0,0\rangle]}\,{>}}{[\langle 0,0\rangle]}\,{>}$$

In many cases, we do not even return to the tonic after our excursion, continuing with a cadence straight after the *IV*. This is the purpose of the backward-facing colouration lexical category (Colour-IVb in figure 2.13) and the semantics ignores the *IV* in the same way.

3.5 Development Semantics

The development combinatory rule combines sequences of tonic passages and resolved cadences into larger units, ultimately into a whole piece of music. Every logical form introduced so far has been a single-item list. The behaviour of the development rule's semantics is rather trivial. It simply concatenates its two arguments: the syntax ensures these are lists.

Example (4) shows a pair of resolved cadences being combined in this way. Example (5) shows a tonic (a single-element list) combining with a subsequent resolved cadence.

(4)
$$\dfrac{\dfrac{\dfrac{\text{IIm}^7}{\lambda x.leftonto(x)} \quad \dfrac{\dfrac{\text{V}^7}{\lambda x.leftonto(x)} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{[lefonto(\langle 0,0\rangle)]}\,{>}}{[leftonto(leftonto(\langle 0,0\rangle))]}\,{>} \quad \dfrac{\dfrac{\text{V}^7}{\lambda x.leftonto(x)} \quad \dfrac{\text{I}}{[\langle 0,0\rangle)]}}{[lefonto(\langle 0,0\rangle)]}\,{>}}{[leftonto(leftonto(\langle 0,0\rangle)), lefonto(\langle 0,0\rangle)]}\,\mathbf{dev}$$

(5)
$$\dfrac{\dfrac{\dfrac{\text{I}}{[\langle 0,0\rangle]} \quad \dfrac{\text{IIm}^7}{\lambda x.leftonto(x)} \quad \dfrac{\dfrac{\text{V}^7}{\lambda x.leftonto(x)} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{[lefonto(\langle 0,0\rangle)]}>}{[leftonto(leftonto(\langle 0,0\rangle))]}>}{[\langle 0,0\rangle, leftonto(leftonto(\langle 0,0\rangle))]}\textbf{dev}$$

3.6 Coordination Semantics

Logical forms representing unresolved cadences can be *coordinated* to share their eventual resolution. This is carried out by the special musical *coordination* combinator. The semantics of this combinator simply conjoins the cadence logical forms using the $\wedge$ operator. Note that, unlike in the logical semantics of natural language, this conjunction operator must preserve the order of its arguments.

$$A \wedge B \not\equiv B \wedge A$$

We can also reduce brackets to reflect the associativity of the conjunction operator.

$$\begin{aligned} A \wedge B \wedge C &\equiv (A \wedge B) \wedge C \\ &\equiv A \wedge (B \wedge C) \end{aligned}$$

$$A \wedge (B \wedge C) \Rightarrow A \wedge B \wedge C$$

$$(A \wedge B) \wedge C \Rightarrow A \wedge B \wedge C$$

The functions that denote cadences are simply conjoined by $\wedge$:

(6)
$$\dfrac{\dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.leftonto(leftonto(x))} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.leftonto(leftonto(x))}}{\lambda x.leftonto(leftonto(x)) \wedge \lambda x.leftonto(leftonto(x))}\&$$

The result is treated as a functor that can be applied to its resolution. It reduces under application to a list in the same way as *leftonto* and *rightonto*. Note that the individual cadences are not actually applied to the resolution at this stage.

(7)
$$\dfrac{\dfrac{\dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.leftonto(leftonto(x))} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.leftonto(leftonto(x))} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{\lambda x.leftonto(leftonto(x)) \wedge \lambda x.leftonto(leftonto(x))}\&}{[(\lambda x.leftonto(leftonto(x)) \wedge \lambda x.leftonto(leftonto(x)))(\langle 0,0\rangle)]}>$$

More than two cadences can be coordinated to share the same resolution. (The predicate *leftonto* is abbreviated to L to save space.)

(8)
$$\dfrac{\dfrac{\dfrac{\dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{\lambda x.L(L(x)) \wedge \lambda x.L(L(x))}\&}{\lambda x.L(L(x)) \wedge \lambda x.L(L(x)) \wedge \lambda x.L(L(x))}\&}{[(\lambda x.L(L(x)) \wedge \lambda x.L(L(x)) \wedge \lambda x.L(L(x)))(\langle 0,0\rangle)]}>$$

The result of a coordination (once applied to its resolution) can become the recursive resolution of a prior cadence step.

(9)
$$\dfrac{\dfrac{\dfrac{\text{VI}^7}{\lambda x.L(x)} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))}}{\dfrac{\lambda x.L(L(x)) \wedge \lambda x.L(L(x))}{[(\lambda x.L(L(x)) \wedge \lambda x.L(L(x)))(\langle 0,0\rangle)]}\,>} \quad \dfrac{\dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{\lambda x.L(L(x)) \wedge \lambda x.L(L(x))}\,\&}{[L((\lambda x.L(L(x)) \wedge \lambda x.L(L(x)))(\langle 0,0\rangle))]}\,>$$

However, this logical form will result in the same tonal space path as that which would have been produced by composing the VI^7 with the following $\text{IIm}^7\ \text{V}^7$ before coordinating:

(10)
$$\dfrac{\dfrac{\dfrac{\dfrac{\text{VI}^7}{\lambda x.L(x)} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))}}{\lambda x.L(L(L(x)))}\,\mathbf{B}} \quad \dfrac{\text{IIm}^7\ \text{V}^7}{\lambda x.L(L(x))} \quad \dfrac{\text{I}}{[\langle 0,0\rangle]}}{\lambda x.L(L(L(x))) \wedge \lambda x.L(L(x))}\,\&}{[(\lambda x.L(L(L(x))) \wedge \lambda x.L(L(x)))(\langle 0,0\rangle)]}\,>$$

$$[(\lambda x.L(L(L(x))) \wedge \lambda x.L(L(x)))(\langle 0,0\rangle)]$$

We therefore define the following equivalence in the logical forms and by convention reduce the left-hand side form to the right-hand side wherever possible.

$$A((B \wedge ...)(C)) \Rightarrow (A \circ B \wedge ...)(C)$$

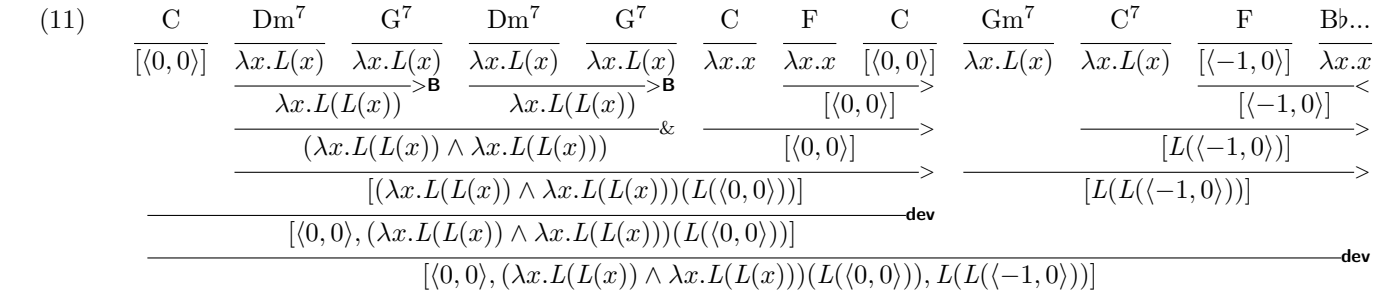
3.7 Bigger Examples

To show all of these things in action at once, example (11), which I have invented for the purpose, but which is a plausible jazz chord sequence, shows the compositional semantics being combined for a longer sequence. The analysis represented by this interpretation may be expressed verbally as follows:

> The sequence begins on a tonic, cadences back to the same tonic, with a *I IV I* colouration on the tonic. It then modulates, via a cadence, to a new key – the subdominant key. When it gets to its new tonic, there is another *IV* colouration, this time not returning to the tonic.

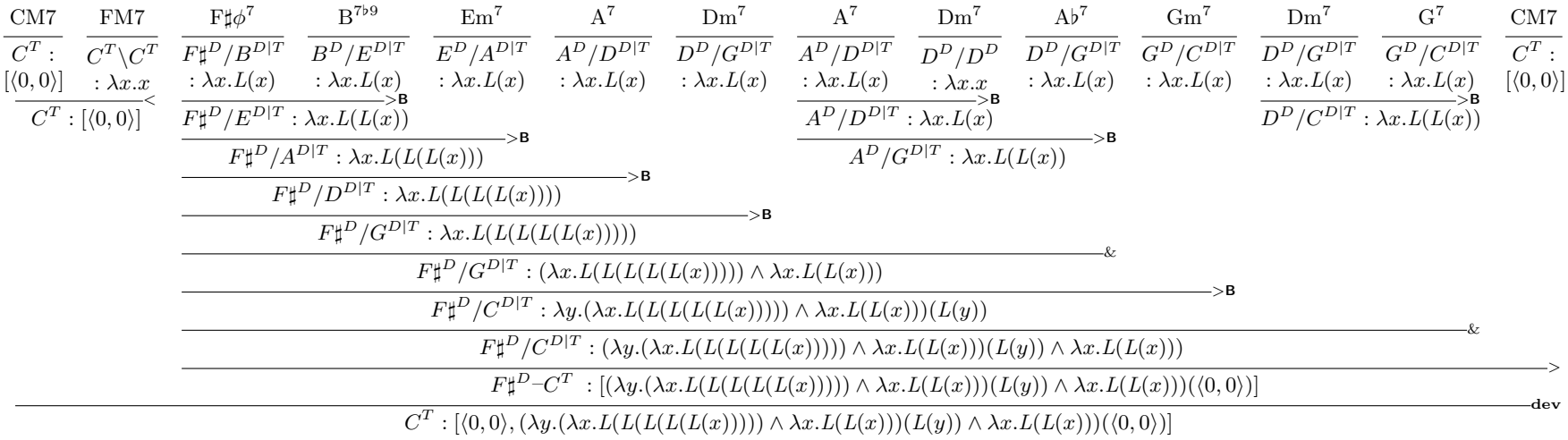
The predicate *leftonto* is replaced by L to conserve space.

(11)



Example (11) shows only the logical forms that are associated with syntactic types during derivation. For a full, real life example, example (12) gives the derivation of a long extended cadence from *Alice in Wonderland*, including logical forms. This is the same derivation that was shown without its semantics in figure 2.15.

(12)



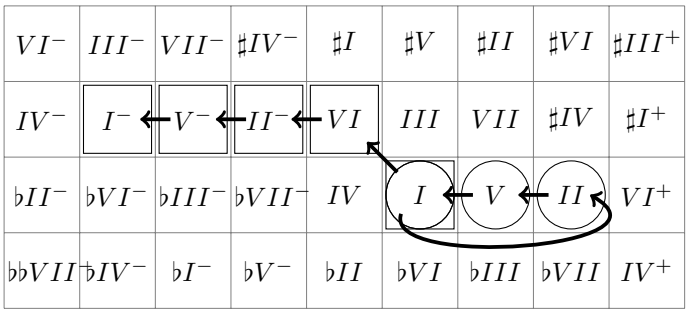


Figure 3.2: The tonal space paths corresponding to two logical forms. $[\langle 0,0\rangle, \mathit{leftonto}(\mathit{leftonto}(\langle 0,0\rangle))]$ (circles) begins at I, $(0,0)$, jumps to II, $(2,0)$, and left-steps back to I. $[\langle 0,0\rangle, \mathit{leftonto}(\mathit{leftonto}(\mathit{leftonto}(\langle 0,0\rangle)))]$ (squares) also begins at I, but jumps to VI, $(-1,1)$, and left-steps to I^-, $(-4,1)$.

3.8 Extracting the Tonal Space Path

The logical forms that come out of the above semantics represent certain constraints on paths through the tonal space. Although the tonic points are ambiguous in the representation, we need only specify one further implicit constraint for all the points of a path to be unambiguous, modulo an arbitrary choice of starting point. Given the fully-specified position of the first point on the path, the rest of the points are constrained to individual points on the tonal space.

Let us first examine the constraints encoded in the various types of predicate. The most obvious constraint is on the point created by a left (or right) movement, denoted in the semantics by *leftonto* (or *rightonto*) predicates. In $\mathit{leftonto}(p)$, the point at which the movement begins must be one step in the grid to the right of the first point of the path p. If the point (x,y) if fully specified, the whole path $\mathit{leftonto}(\mathit{leftonto}((x,y)))$ is therefore also unambiguous.

Two cadences that share a resolution through coordination are constrained to end at the same point, since their points are constrained relative to their resolution.

There is no obvious constraint between items in the top-level list of tonics and resolved cadences and this is where we must add a further implicit constraint. The most plausible choice of relative positions is reached by constraining the start point of a particular item in the list to be the closest possible point that satisfies all other constraints to the end point of the previous item.

For example, take the following two logical forms:

1. $[\langle 0,0\rangle, \mathit{leftonto}(\mathit{leftonto}(\langle 0,0\rangle))]$
2. $[\langle 0,0\rangle, \mathit{leftonto}(\mathit{leftonto}(\mathit{leftonto}(\langle 0,0\rangle)))]$

The tonal space paths for these logical forms are shown in figure 3.2.

The start of the second item in path 1 is dependent, ultimately, on the cadence resolution $\langle 0,0\rangle$. But this point is underspecified: we can choose for it any of the infinite points that lie at $\langle 0,0\rangle$ within their enharmonic block. Given an arbitrary choice of the first item's point at the central $(0,0)$, we will choose the same point for the end of the second item, since it puts the start of the second item (now $(2,0)$) as close as possible to $(0,0)$. A choice of $(-4,1)$ for

the end point would also have been permitted by other constraints, but would have resulted in a larger jump between the two path fragments.

In path 2, however, the second path begins at a point further from its ending. In this case we will choose $(-1, 1)$ as the start point for the second item by setting the $\langle 0, 0 \rangle$ at its end to be at $(-4, 1)$.

Note that the choice of the first point on the path is unimportant: two paths identical in form, but occurring at different positions in the space can be considered equivalent, since the only difference between them is their absolute pitch and we (uncontraversially) consider precise absolute pitch not to be pertinent to musical semantics. In both the above examples, we could have chosen $(4, -1)$, for instance, as the coordinate $\langle 0, 0 \rangle$ at the start and the resulting paths would be considered identical to those we derived.

A simple algorithm can be constructed by means of a recursive transformation of the logical predicates to produce the flat tonal space path represented by a logical form generated by the grammar.

As well as demonstrating that any logical form is interpretable as an analysis in the tonal space, there are circumstances in which this transformation is of use. For example, in chapter 2 we use path similarity between an output interpretation and the gold standard as an evaluation metric. The paths we compare are those produced by this algorithm.

Chapter 4

Extending the Parser for Analysis of Musical Notes

4.1 Introduction

Raphael & Stoddard (2004) proposed a model for shallow harmonic analysis of MIDI data. The model assumes a division of the data into bars, or similar units. It assigns to each bar a label consisting of three parts: *tonic*, *mode* and *chord*. The tonic is a choice of chromatic pitch class, which together with the mode (major or minor) defines the key. The chord is a selection from the seven possible chord roots in that key. So, for example, a chord may be analysed as a V chord in the key of A major. This implies the chord label E^7, but tells us more than just this chord label.

My interest in a model such as this is in its potential use as the basis for a sequence model to assign grammatical categories to MIDI data. Some of the information captured by the analyses of the Raphael and Stoddard model overlaps with that information required to make the decision as to which category to assign to a passage of MIDI data. My eventual goal is to replace our current supertagger component, which assigns categories from the jazz lexicon to chords in a chord sequence, with a supertagger for MIDI data; this would partition the data into chords and assign a category to each, which would serve as input to the parser.

The first questions to answer are whether the Raphael and Stoddard (henceforth *R&S*) model can be applied to jazz standard MIDI files and successfully learn the same sort of information it is able to learn from classical music; and whether the model's analyses are sufficiently accurate that they could serve as a good starting point for a MIDI supertagging model. Then I can address the question of how such a supertagging model might be constructed.

The first task necessary prior to addressing these questions is to replicate the original R&S model, training on their data and producing the results they report. In section 4.2, I give an overview of the R&S model. In section 4.3, I discuss the problems involved in reimplementing the original model according to the description in the paper. I discuss some particular problems with naïvely applying the same model to the jazz domain in section 4.4.

In section 4.5 I outline some experiments I have performed with my imple-

mentation of the R&S model, the solutions I adopted to the problems of section 4.3 and the datasets I used. Raphael & Stoddard (2004) present no empirical evaluation of their model, but describe some examples of its output. I describe some examples of the model's performance on different kinds of data in section 4.6.

Finally, I present some thoughts on the adaptation of the R&S model to the supertagging problem and propose some possible model structures in section 4.8.

4.2 Model Overview

The model of Raphael & Stoddard (2004) is a hidden Markov model (HMM), with state labels representing harmonic analysis, emitting the observed notes in each time period. A fixed time period is used, which is the minimum period to which a single harmonic label may be assigned. This is set to be one bar in most cases, half a bar in others. It is required that the division of the note sequence into bar units is encoded in the MIDI data.

The set of state labels is:

$$L = T \times M \times C = \{0, \ldots, 11\} \times \{\mathit{major}, \mathit{minor}\} \times \{I, \mathit{II}, \ldots, \mathit{VII}\}$$

That is, a state label represents a choice of a tonic pitch class (C=0, D♭=C♯=1, etc), a mode and a chord in the key defined by the tonic and mode. The chord is chosen from the seven triads built on the degrees of the scale. A further chord – the dominant seventh, which includes the dominant seventh note among the notes considered to be a part of the chord – is added as something of an afterthought. The authors mention that other such chords could also be added to the vocabulary, but content themselves with these eight for the purposes of their experiments.

The set of states thus becomes:

$$L = T \times M \times C = \{0, \ldots, 11\} \times \{\mathit{major}, \mathit{minor}\} \times \{I, \mathit{II}, \ldots, \mathit{VII}, \mathit{V7}\}$$

The authors claim that secondary functionality, such as the secondary dominant D^7 in D^7 G^7 CM7, is handled as a modulation. In other words, the D^7 would be interpreted as a V chord in the key of G major, then the G^7 as a V in the key of C major.

The notes of a particular time segment (bar) are treated as a bag of pitch classes, with only certain aspects of their timing within the segment and no information about their voicing playing a role in the model. The form of the emission distribution is described below.

Parameter tying in the transition and emission distributions allows the model to ignore absolute pitch and severely reduces the number of parameters, making it feasible to train the model using the Baum-Welch form of the expectation-maximization (EM) algorithm. The model is trained on unlabelled data: MIDI data with no annotations of harmonic analysis. The model is biased to give a sensible interpretation to the labels by initializing the parameters of the emission distribution so that it begins by naïvely recognizing the basic triads of each chord as highly probable notes. The authors claim that initialization of the transition distribution has little effect on the trained model.

4.2.1 Transition Distribution

A series of simplifying assumptions is made to reduce the number of parameters to the model. I stepp through these below. A state is represented as a triple $(t, m, c) \in L$.

We assume that the choice of t' and m', the key labels, is not dependent on the previous chord label:

$$\begin{aligned} p(x'|x) = &\quad p(t', m', c'|t, m, c) \\ = &\quad p(t', m'|t, m) \cdot p(c'|t', m', t, m, c) \end{aligned}$$

Next, the choice of new tonic pitch class is expressed relative to the previous tonic. This makes the key transition distribution ignorant of absolute pitch distinctions, since parameters are now shared across all key transitions that have the same interval between their tonics.

$$p(x'|x) = \quad p(t' - t, m'|m) \cdot p(c'|t' - t, m', m, c)$$

We further assume that the choice of chord (within a key) does not depend on the current key, but only on the previous chord – another aspect of relative pitch. The choice of chord is also not affected by mode: it is reasonable to expect the chord transition probabilities to be similar in different modes, so this is not a very strong assumption.

$$p(x'|x) = \quad p(t' - t, m'|m) \cdot p(c'|c)$$

Finally, we ignore the previous chord in choosing the chord whenever the key (tonic or mode) is changing. The first chord after a key change is chosen from a separate distribution.

$$p(x'|x) = \quad p(t' - t, m'|m) \cdot \begin{cases} p(c'|c) & t' = t, m' = m, \\ p(c) & \textit{otherwise} \end{cases}$$

The parameters of each of these three component distributions can be estimated independently by counting transitions in the data (or using pseudocounts in EM).

4.2.2 Emission Distribution

The each note in a segment is emitted independently, with a probability depending on the state label and the time in the segment at which it occurs. The probability of a segment is the product of the probabilities of its notes.

$$\begin{aligned} p(y|x, r) = &\quad p(y^1, \ldots, y^K|x, r^1, \ldots, r^K) \\ = &\quad \textstyle\prod_{k=1}^{K} p(y^k|x, r^k) \end{aligned}$$

The probabilities depend on the onset time only in a limited manner. The vector y contains the pitch class of each note in a segment. The vector r contains information about each note as follows:

$$r^k = \begin{cases} 0 & y^k \textit{ occurs at start of bar} \\ 1 & y^k \textit{ occurs in middle of bar (beat 3)} \\ 2 & y^k \textit{ occurs on offbeat (2 or 4)} \\ 3 & \textit{otherwise} \end{cases}$$

Instead of estimating probabilities for each pitch class that could be emitted from each state, the probability of a note is estimated on the basis of which of five classes it falls into in relation to the chord and scale denoted by the state label: root, third or fifth of the chord, another note of the scale, or a non-scale note.

$$p(y^k|x,r) = \frac{p(d(y^k,x)|r^k)}{V(d(y^k,x))}$$

where

$$d(y^k,x) = d(y^k,t,m,c) = \begin{cases} 1 & y^k \textit{ is root of chord } t,m,c \\ 2 & y^k \textit{ is 3rd of chord } t,m,c \\ 3 & y^k \textit{ is 5th of chord } t,m,c \\ 4 & y^k \textit{ is in scale } t,m \\ 5 & \textit{otherwise} \end{cases}$$

and

$$V(1) = V(2) = V(3) = 1 \qquad V(4) = 4 \qquad V(5) = 5$$

Presumably d is extended for the addition of chords larger than triads, such as the dominant seventh chord, but this is not mentioned. The simplest extension would be to add the additional notes into one of the existing classes: for example, to alter the definition so that $d = 3$ *if* y^k *is 5th or 7th of chord* t,m,c.

4.2.3 Training

No annotated data is used to train the model. Instead it is trained using the *Baum-Welch algorithm*, an instance of EM for unsupervised training of HMMs. The model is biased towards learning a sensible model by initializing the emission distribution parameters to reflect a simplified form of the distribution that we would expect each type of chord to have.

The parameters are set initially such that the notes most likely to be emitted from a state are the notes of the triad denoted by the state label. The notes of the state's scale are less probable and non-scale notes least probable. These initial parameters are not dependent on r^k, the rhythmic position.

The intuition is that from the first EM iteration, the model will be able to recognise certain state labels as being highly probable for bars that contain primarily the notes of the triad and notes of the scale. As the parameters converge, the model should learn to recognise a more realistic distribution of notes from each state.

The authors state that initialization of the transition probabilities makes little difference to the outcome of the training.

4.3 Difficulties with Replicating the Model

A number of details are omitted from the paper which are important to replicating the results reported. I have had to make a decision in each case as to how to implement the model. In this section, I outline the problems and the solutions I have adopted.

4.3.1 Emission Distribution Initialization

The paper does not give **details of the initialization of the emission distribution**. It only says that "if a particular chord and key are sounding, then the members of that chord should be the most likely pitches, the other notes in the key should be somewhat less likely, while the pitches not in the key are the least likely." It would seem from this that the three notes of the triad ($d \in \{1, 2, 3\}$) are initialized to be equally probable and to have a greater total probability than other scale notes ($d = 4$). Non-scale notes ($d = 5$) receive a smaller value. It is unlikely that the precise values given to these parameters are important, provided these constraints are satisfied.

I will refer to the total triad probability as d^0_{triad}, the scale note probability as d^0_{scale} and the non-scale note probability as d^0_{other}. The above constraints may be expressed as:

$$d^0_{triad} > d^0_{scale} > d^0_{other}$$

In my experiments, I have set these parameters as follows:

d^0_{triad}	0.5
d^0_{scale}	0.3
d^0_{other}	0.2

4.3.2 Transition Distribution Parameters

The authors claim that certain parameters of the transition distribution are not learnt well by the training procedure. They set these parameters by hand at the end of the training. They do not say which parameters these are or what values they set them to. Naturally, this is problematic for replicating the behaviour of their model.

One solution to this is simply to look at how the model performs without hand-setting of parameters. Another is to look at the trained parameters and use my own intuitions about what we expect to see and adjust them as appropriate.

For the time being, I have adopted the first of these. We must bear in mind, therefore, that the model could probably perform better with some manual tweaking of parameters, but I see little point in trying to guess which parameters the authors chose to tweak in their model and how.

4.3.3 Datasets

The paper reports no empirical results of testing the model. This is because the problem of harmonic analysis is somewhat subjective. The authors expect any human annotation of the data to be unreliable, since annotator agreement would be very low, so that it would be meaningless to use such an annotation to judge the level of correctness in a model's output.

They do have some examples of the output of the model (Christopher Raphael, PC), in the form of automatically annotated MIDI files. One way of verifying that the model is doing roughly what it should be doing is to compare its output on these MIDI files to the annotations made by their implementation and check that they are similar.

None of the original data used to train the model is available. We know that it consisted of several movements of Haydn piano sonatas, similar to one of the annotated example output files available, but more than this is lost in the mists of time (Christopher Raphael, PC).

In order to have a dataset similar to that used by the authors to train the models on, I have collected several MIDI files from the web of Haydn piano sonatas. These are very similar to the Haydn MIDI file available as an example of the system's performance. To be precise, I have used MIDI encodings of the following works by Haydn:

- Piano Sonata 35, movement 1
- Piano Sonata 35, movement 2
- Piano Sonata 40, movement 2
- Piano Sonata 31, movement 1
- Piano Sonata 31, movement 3

These are all the movements from sonatas 31, 35 and 40, excluding those in triple time (which cannot be handled by the R&S models).

4.4 Adapting to Jazz Data

For several reasons, it is not obvious that the original R&S model can be applied directly to jazz data with any success. I discuss several musical differences here that could cause problems. Later we will see to what extent these speculations are born out when the model is trained.

4.4.1 Cadences

Extended cadences, with many levels of secondary dominant, or sometimes subdominant, function, are common in jazz, but do not play such an important role in the domains of the original model. Such secondary function, the authors claim, should be handled by the model as modulation combined with dominant (or subdominant) function.

Examination of the parameters of the model trained on Haydn data (section 4.5.1, parameters in appendix A.1) suggests that this may not be consistently the case. The distribution of chords given a previous chord V is topped by I, suggesting that the dominant function has been learned. Given a previous chord II, the most probable chord is V (0.47), suggesting that it often serves as a secondary dominant function. With previous chord VI, the most probable chord is II (0.56). VI appears to be used rarely, but then usually as a dominant to a II. The distribution of chords following III, however, shows no sign of dominant function. Unsurprisingly, key transitions to the relative IV and V are highly probable, which would also permit interpretation of secondary function by modulation.

All in all, the treatment of extended cadences by the model is unclear and possibly mixes the two approaches of using certain chord labels as secondary and higher-level dominants and using modulation. This is not helped by the

independence of key transition on previous chord and the independence of chord after a key change of any context. If extended authentic cadences are to be well predicted by the model using modulation, it must be able to encode the strong expectation that a V chord will be followed by modulation to the key of IV, and that the following chord will also be a V. This would not cause a great problem in analyzing, for example, Haydn, but will not be correct for jazz.

4.4.2 Time Segmentation

Syncopation, common in jazz, causes problems with the assumption in the construction of the original model that knowing the timing of the pulse and the meter of a piece on its own allows us to segment the music temporally into passages within which the underlying harmony does not change. In the case of the original data, this assumption was occasionally violated when the harmony changed more quickly than the chosen atomic time period (such as half a bar), but this was rare.

In certain styles of performance in jazz it is very common the preempt a harmonic change slightly (often by a third of a beat). The above segmentation technique will consider these notes played ahead of the beat to have been a part of the previous segment, but a human listener will have no trouble in associating them with the harmony that follows on the next downbeat. In other cases, notes from the established chord may be played in the same proximity before the harmonic change and a listener will not interpret this as syncopation.

In order to handle these and other similar complexities in the rhythm, the model will need a very much more sophisticated metrical model that, together with the harmonic model, can identify the times of harmonic movements and recognise notes that deliberately fall just outside these boundaries.

There is no simple way to incorporate a metrical model like this into the R&S model. The problem calls for a model that identifies metrical and harmonic structures simultaneously, as does the (non-probabilistic) model of Lerdahl & Jackendoff (1996).

4.4.3 Chord Voicing

The construction of the R&S model is based on the observation that chords are identified primarily by notes of a triad (or a tetrad), then by the notes of the chord's associated scale. The training process relies on being able to identify chords to a least a small level of accuracy with only a rough estimate of the probabilities of each of these notes occuring in a passage which the chord underlies. If this accuracy is sufficiently high, the model is able to learn a better distribution, though still constrained to be expressed in terms of the probability of observing the notes of the triad and the notes of the scale.

This approach is fairly successful applied to data from the original domain. Chords and keys can indeed be well identified by their triads and scales. Voicing of chords in jazz is more varied and consequently typically more ambiguous. In any domain, notes outside the chord's triad are common in melody lines, but in jazz accompanying instruments, often primarily serving the purpose of supplying the notes of the harmony, use very different sets of notes to do this. The difference may be seen by examining some of the typical chord voicings used by such, for example, a pianist.

In jazz, the triad of a chord is less important. The third is usually expressed, but the fifth is often omitted and it is not uncommon for even the root to be unexpressed. At the same time, the seventh is very much more important, the major seventh commonly used in constrast to the dominant seventh. Further additions to chord notes are also common. Ninths are used frequently in chords of any function, whilst a flat ninth (an out-of-scale note for R&S) suggest dominant function. Elevenths and thirteenths and often added to dominant chords.

The emission distribution of the R&S model is unable to capture the different probabilities of these notes in different chords, since they are not among those distinguished by the d function.

Despite this huge ambiguity, a listener is able to identify the harmonic structure of cadences. One helpful clue is in the ordering and octaves of the pitches. Certain notes are more likely to appear in high or low octaves: the root is likely to appear in the bass, whilst a thirteenth is much less likely. These distinctions are lost in the pitch class representation, so cannot be captured by the model.

Another clue is in voice leading. In classical music, the descending chromatic scale resulting from a series of chords descending in perfect fifths (the line alternating between the major third and dominant seventh degrees of each chord) has been thoroughly used by composers. In jazz, it is even more common to express the dominant seventh note in a dominant chord, so this pattern is easily recognised and characteristic of a cadence. Such patterns are not captured by the R&S model, though they propose a structure of an alternative model that does attempt to incorporate voice leading.

Voicing of chords also presents a metrical problem similar to that discussed above as making segmentation difficult. Typically certain instruments maintain the meter (often drums and bass, though this may vary), whilst others expressing the harmony have great metrical freedom. It may often be the case that the majority of chord notes are played on weak beats or off the beat. The rhythmic component of the R&S emission distribution is based on the observation that the chord notes are more likely to be expressed at certain points in the metrical structure (namely on stronger beats), or at least that the probabilities of the different classes of notes depend on the beats they occur on. This would seem less likely to hold strongly for jazz music. I comment further on this in section 4.6.1.

4.5 Experiments

In this section, I describe each of the models that I have trained, beginning with a replication of the original R&S model, followed by several variants. For the complete set of parameters that resulted from training each model, see appendices A.1–A.4.

4.5.1 Haydn Model

My first experiments were an attempt to replicate the original R&S model as closely as possible (with the exception of hand-setting of transition parameters, as discussed above). I trained the model HAYDN using Baum-Welch on five Haydn piano sonata movements. For this model, I used only the triads on the

seven scale degress as the set of available chords (the chord component of the state set definition).

4.5.2 Adding Dominant Sevenths

I trained a second model HAYDN7, identical to HAYDN, but this time, as in the experiments referred to in Raphael & Stoddard (2004), using a set of chords made up of each scale degree's triad, plus the dominant seventh chord (tetrad). I adopted the approach I proposed in section 4.2.2 to do this.

4.5.3 Jazz Model

I trained a model JAZZ, identical in structure to HAYDN, using jazz MIDI data. The dataset was a subset of my own corpus of jazz MIDI files, collected from the web. I manually added the information needed by the R&S model to each of a small subset (12 files) of the corpus – the shortest time period to allow chords to occupy and a temporal offset value to ensure these periods coincide with bar lines.

4.5.4 Jazz with Dominant Sevenths

Just as with HAYDN, I trained a model like JAZZ, but now using a chord set including the tetrad in addition to the triad on each scale degree. I will refer to this as JAZZ7.

4.5.5 Initializing the Transition Distribution

The authors claim that initialization of the chord function transition distribution makes little or no difference to the outcome of training. This is a claim worth testing, since intuitively it seems as though biasing the model towards an understanding of the non-tonic function chords in terms of their resolution would be an important part of training the model. We can also not be sure that if this claim holds for the original training data it will necessarily hold for our jazz training data, which exhibits a more complex notion of chord function through extended cadences.

I trained a model JAZZFUN identical to JAZZ, but with the following parameters of the chord transition distribution set at initialization. Remaining probability mass is distributed uniformly.

Transition	Probability
II→ II	0.2
II→ V	0.8
III→ III	0.2
III→ VI	0.8
IV→ IV	0.2
IV→ I	0.8
V→ V	0.2
V→ I	0.8
VI→ VI	0.2
VI→ II	0.8
VII→ VII	0.2
VII→ III	0.8

This should have the effect of biasing chords V, II, VI, III and VII towards a dominant interpretation and IV towards a subdominant interpretation.

4.5.6 Unigram Baseline

In order to assess what is being acheived by the state transition probabilities (the key change distribution, chord transition distribution and key-initial chord distribution), I have implemented a unigram version of the model JazzUnigram. This is structured identically to the previous models, but with the transition distributions set to uniform distributions and not updated during EM training. This is equivalent to a unigram model, in which the probability is estimated of a chord given each possible state label, irrespective of surrounding state labels or chords. I trained the emission distributions using Baum-Welch as before.

4.6 Model Analysis

The parameters of the distributions of the model can each be fairly intuitively interpreted with respect to high-level characteristics of the model. In this section, I examine each of the distributions in turn.

The emission distributions represent the relative expectation of the notes of a chord in different rhythmic positions.

The chord transition distributions represent, in the case of non-tonic chords, the probability of a chord's resolution to other possible following chords. This is not strictly reflected in the distribution, since noise is added by cases where the chord does not resolve, such as cadence coordination. However, we would hope to see, for example, that a V chord has a high probability of resolving to a I. It may also continue to any other chord, if it occurs at the end of a non-final coordinated constituent, or if its function is incorrectly interpreted, but we can gain some insight into the model's success by observing where the probabilities coincide with the theoretical functional interpretations of the chords.

The key transition distributions partially represent the probabilities of relative key changes. This is slightly diluted by the claim that the model is expected to handle extended cadences as modulations, but as it happens the "key changes" used in these cases would be expected to be highly probable in a pure model of key changes in any case (namely modulation to the relative IV or V).

The interpretation of the post-key change chord distribution is slightly less clear. We would expect that if the model is indeed using modulation to interpret extended cadences, beginning on a V chord after a modulation would be highly probable. Perhaps we would expect I also to have a high probability, since a tonic may serve to establish a new key. Beyond this, though, it is difficult to give any clear interpretation to the probabilities, so I will not discuss them any further here.

4.6.1 Emission Distributions

In every model, every rhythmic position gives a low probability to non-scale notes. The jazz models tend to give a slightly higher probability non-scale notes in weak rhythmic positions than the Haydn models. This is unsurprising, since jazz music generally uses more chromatic notes than Haydn, who keeps more to notes closely related to the chord and tonality. It may also be due to noise introduced by syncopation, which the model does not account for, causing some notes to be contributing to the distribution of the wrong chord. A model with a more sophisticated rhythmic component might be able to distinguish chromatic notes to a large degree on the basis of rhythmic prominence and other factors.

In all models, scale notes (that is, notes outside the triad, but within the scale) are more probable on weaker beats, whilst triad notes are overall more common on strong beats than weak beats. In the jazz models this effect is seen less strongly, scale notes getting a higher probability on strong beats than they do in the Haydn models. This difference may be accounted for by the reasons I suggested above for the higher probability the jazz models give to non-scale notes. Although the effect is weaker in the jazz models, it is worth noting that across all models scale notes are strictly more probable on weak beats or off the beat than they are on the first or third beats. This suggests that even on jazz music, for which many of the assumptions regarding rhythmic prominence made for the original R&S model are violated, the rhythmic division of emission distribution is still somewhat meaningful.

The distribution over the three triad notes varies rather confusingly across rhythmic positions in all models. Surprisingly, the third tends to have low probability in strong rhythmic positions. In all the jazz models, it has a lower probability (though not always by a large margin) than the fifth on the first and third beats. Jazz theory would suggest that the third would be a stronger predictor of the chord than the fifth. However, jazz theory also places great importance on the seventh, which is not accounted for in the design of these models. I discuss the importance of incorporating this into a model of jazz further in section 4.6.3.

4.6.2 Key Transition Distributions

Let us first look at the transition distributions of the Haydn-trained models. As we would hope, HADYN has a strong inclination to remain in the same key: only 10% of transitions from a major key modulate. This is weaker when starting in a minor key, a pattern seen in all the models. This could be a reflection of the greater stability of major keys, but is more likely simply due to the fact that the training data is predominantly major. A large amount of the training data for the minor distribution is likely to be in fact short-term modulations or

extended cadences, in which cases we would expect no strong bias to remain in the same key for long.

HAYDN bears out some of our expectations for a distribution that reflects genuine modulations. The most common keys to modulate to are those most closely related. In a major key these are the relative minor (VIm) and the dominant and subdominant keys (V and IV). In a minor key they are again dominant and subdominant keys, though this time minor (Vm and IVm). Also high up are the dominant and subdominant keys of the relative major key (♭VII and ♭VI).

It is surprising, however, that the relative major key itself (♭III) has such a low probability, appearing only 12th in the list. It is also surprising that ♭VII is as probable as it is, featuring at second place. This latter could be a result of misinterpretation of cadences, treating a II V I cadence (or IV V I, which is more common in this domain) as a modulation for the key of II to I. Both, however, are probably largely a reflection of the lack of genuine minor data, meaning that the probabilities are strongly influenced by noise as a result of misinterpretation in the EM training process.

HAYDN7's distributions are very similar to those of HAYDN, suggesting that the addition of the dominant seventh chord to the model's palette changes the chord transition distributions, but does not greatly affect the model's key analyses, a somewhat comforting result.

So to the models trained on jazz data. JAZZ and JAZZFUN are considerably more inclined to modulate the the Haydn models, and JAZZ7 even more so. This is not surprising, firstly since the music probably genuinely does change key more frequently, and secondly since we expect the model to treat extended cadences (at least in some cases) as modulations. It may, however, make us wonder whether the model is sufficiently well-equiped to deal with this new domain: the conflation of extended cadences and genuine modulations much weakens the usefulness of these statistics, which in the Haydn models are intended to correspond most of the time to a musician's analysis of key changes.

JAZZ's distributions bear many similarities to HAYDN's, with some notable exceptions. The very high probability of modulation from a major key to its IIIm is unexpected: this is a fairly closely-related key, but should not appear as the most probable key to modulate to. The very low probability of modulation to V, and Vm in the minor distribution, is also unexpected and is most likely simply to be a sign of poor quality interpretation of key.

Interestingly, the situation in this respect is much improved in JAZZ7's major distribution. The addition of the dominant seventh chord, in contrast to the Haydn models, has affected the key interpretation, and seemingly for the better. Unfortunately, however, the minor distribution is further diluted, though it displays a similar ordering to that of JAZZ.

Finally, the distributions of JAZZFUN are very similar to JAZZ. Recall that the only difference between these models was the initialization of their transition distributions. This seems to corroborate the authors' surprising claim that initialization of the transition distributions makes little or no difference to the outcome of the training.

4.6.3 Example Output

The authors decline to perform any form of evaluation of their model's performance on the basis that it is impossible to define a gold standard harmonic analysis. It is true that any particular sequence may be analysed in multiple ways, each equally correct. A possible form of evaluation they do not mention is to have experts mark the analyses output by the model as valid or invalid, allowing for the possibility that there may be multiple valid analyses and also that there could be an analysis of one chord whose validity is contingent on its surrounding chords having a compatible interpretation. This is, of course, time consuming and labour intensive, hence the use of pre-annotated gold standards where possible.

I do not attempt here such an evaluation. Instead I present the output of several of the models applied to the same MIDI file side by side: *Can't Help Lovin' Dat Man*. The table shows the harmonic analysis assigned at each time point by each model and also the chord label implied by this analysis. I do not here evaluate the validity of the harmonic labels, but just the chord labels, which can be (and often are) correct even where the analysis is mistaken.

In the final column are my own "gold standard" chord labels, which I have transcribed by examining the notes of the bar and by listening to the MIDI file. There are several cases where the granularity chosen for the labelling is not sufficiently fine – where two chords occur in the same bar[1].

I have marked in bold those automatic chord labels which I consider to be incorrect. This process warrants some explanation. There are many cases in which the chord label assigned by the model is not identical to that of the gold standard, but where I have marked it as correct. Sometimes this is because one chord is simply an inversion of the other (such as C6 vs. Am^7): often, such as in the case of diminished chords, the only reason for choosing one over the other is the note that is played in the bass – information to which the model does not have access.

In some cases, the precise chord label may be freely chosen depending on ease of voicing or transcription policy. For example, many dominant seventh chords contain high additions (11, 13, etc) and may reasonably be considered diminished seventh chords, provided one of the four possible correct roots is used.

Only one of the models, Jazz7, includes the dominant seventh chord, so is able to assigned the chord label X^7. To make a fair comparison, since I am not evaluating the choice of functional label for other models, I have not penalised this model for using the dominant seventh label where it is incorrect.

The interest in this exercise lies in the comparison of the mistakes made by the different models, rather than the overall number of mistakes, since it is such a small amount of data and the marking procedure involves the fairly arbitrary policies described above. For a more objective evaluation, these would need to be better motivated and defined and, of course, more output would need to be examined. It is therefore important not to take too seriously the numbers at the bottom of the columns, which denote the total number of chords marked correct.

[1] I use the term "bar" here to refer to the smallest time unit to which the model may assign a chord. In fact, the size of this unit is usually set to what would normally be transcribed as half a bar.

Model Output

Bar	Haydn		Jazz		Jazz7		JazzFun		JazzUnigram		Manual
2	E♭ major: I	E♭	E♭ major: V	**B♭**	A♭ major: V^7	$E♭^7$	A♭ major: V	E♭	B♭ major: IV	E♭	E♭
3		**E♭**	E♭ major: I	**E♭**	D♭ major: V^7	$\mathbf{A♭}^7$	D♭ major: II	**E♭m**	B major: III	**E♭m**	$G♭^7$
4		**E♭**		**E♭**	E♭ major: V^7	$\mathbf{B♭}^7$		**E♭m**	A♭ major: V	**E♭**	Fm^7
5	E♭ major: V	**B♭**	E♭ major: VII	**Ddim**		$\mathbf{B♭}^7$	E♭ minor: V	**B♭**	A minor: VII	A♭dim	E^7
		B♭		**Ddim**		$\mathbf{B♭}^7$		**B♭**	B♭ major: I	**B♭**	Gm^7
7	E♭ major: II	**Fm**		**Ddim**		$\mathbf{B♭}^7$	E♭ minor: I	E♭m	E♭ minor: VII	**Ddim**	E♭m
8		Fm		**Ddim**		$\mathbf{B♭}^7$	E♭ major: II	Fm	A♭ major: VI	Fm	Fsus4
9	E♭ major: V	**B♭**	E♭ major: III	**Gm**	C minor: V^7	$\mathbf{G}^7$	E♭ major: V	**B♭**	B minor: VI	**G**	Em^7 (13)
10	E♭ major: I	E♭	E♭ major: I	E♭	A♭ major: V^7	$E♭^7$	E♭ major: I	E♭	A♭ major: V	E♭	E♭6
11	A♭ minor: II	B♭dim	B major: V	G♭	B major: V^7	$G♭^7$	B major: V	G♭	B major: V	G♭	$G♭^7$
12	E♭ major: II	Fm	E♭ major: II	Fm	E♭ major: V^7	$\mathbf{B♭}^7$	E♭ major: II	Fm	E♭ major: II	Fm	Fm^7 (11)
13	E♭ major: V	B♭	E♭ major: V	B♭		$B♭^7$	E♭ major: V	B♭	E♭ major: V	B♭	$B♭^7$
14	E♭ major: I	E♭	E♭ major: I	E♭	A♭ major: V^7	$E♭^7$	E♭ major: I	E♭	B♭ major: IV	E♭	$E♭^7$
15	C minor: I	**Cm**	C minor: VI	A♭	D♭ major: V^7	$A♭^7$	C minor: VI	A♭	D♭ minor: V	A♭	$A♭^7$
16	C minor: IV	Fm		A♭		$A♭^7$		A♭	C minor: VI	A♭	A♭6
17		**Fm**		**A♭**	G♭ major: V^7	$\mathbf{D♭}^7$		**A♭**	G♭ major: II	A♭m	A♭m6
18	B♭ major: IV	E♭	B♭ major: IV	E♭	A♭ major: V^7	$E♭^7$	B♭ major: IV	E♭	B♭ major: IV	E♭	E♭
19	G minor: IV	Cm	G minor: IV	Cm	G minor: V^7	$\mathbf{D}^7$	G minor: IV	Cm	B♭ major: II	Cm	Cm
20	G♭ major: IV	B	E major: V	B	E minor: V^7	B^7	E major: V	B	E minor: V	B	B^7
21	E♭ minor: VII	Ddim	E♭ major: V	B♭	E♭ major: V^7	$B♭^7$	E♭ major: V	B♭	E♭ minor: VII	Ddim	$B♭^7$
22	B♭ major: IV	E♭	E♭ major: I	E♭	E♭ major: I	E♭	E♭ major: I	E♭	B♭ major: IV	E♭	E♭, Gm
23	G minor: I	Gm		**E♭**	G minor: V^7	$\mathbf{D}^7$		**E♭**	G minor: I	Gm	Gm, G♭dim
24	E♭ major: II	Fm	E♭ major: II	Fm	E♭ major: V^7	$\mathbf{B♭}^7$	E♭ major: II	Fm	A♭ major: VI	Fm	Fm^7
25	E♭ major: V	B♭	E♭ major: V	B♭		$B♭^7$	E♭ major: V	B♭	E♭ major: V	B♭	$B♭^7$
26	E♭ major: I	E♭	E♭ major: I	E♭	A♭ major: V^7	$E♭^7$	E♭ major: I	E♭	A♭ major: V	E♭	E♭6
27	A♭ minor: II	B♭dim	B major: III	**E♭m**	B major: V^7	$G♭^7$	B major: III	**E♭m**	B major: III	**E♭m**	$G♭^7$
28	E♭ major: II	Fm	E♭ major: II	Fm	E♭ major: V^7	$\mathbf{B♭}^7$	E♭ major: II	Fm	E♭ major: II	Fm	Fm^7
29	E♭ major: V	B♭	E♭ major: V	B♭		$B♭^7$	E♭ major: V	B♭	E♭ major: V	B♭	$B♭^7$

Continued...

Bar	Haydn		Jazz		Jazz7		JazzFun		JazzUnigram		Manual
30	E♭ major: I	E♭	E♭ major: I	E♭	E♭ major: I	E♭	E♭ major: I	E♭	B♭ major: IV	E♭	$E♭M^7$
31	F minor: V	C	F major: III	Am	D minor: V^7	A^7	F major: III	Am	G major: II	Am	Am^7 (♭11)
32	C minor: IV	Fm	C minor: VI	A♭	D♭ major: V^7	$A♭^7$	C minor: VI	A♭	C minor: VI	A♭	A♭6
33		**Fm**		**A♭**	G♭ major: V^7	$\mathbf{D♭}^7$		**A♭**	G♭ major: II	A♭m	A♭m6
34	B♭ major: IV	E♭	B♭ major: IV	E♭	A♭ major: V^7	$E♭^7$	B♭ major: IV	E♭	B♭ major: IV	E♭	E♭
35	G minor: IV	Cm	G minor: IV	Cm	G minor: IV	Cm	G minor: IV	Cm	B♭ major: II	Cm	Cm^7
36	G♭ major: IV	B	E major: V	B	E major: V^7	B^7	E major: V	B	E minor: V	B	B^7
37	E♭ minor: V	B♭	E♭ minor: V	B♭	E♭ minor: V^7	$B♭^7$	E♭ minor: VII	Ddim	E♭ minor: VII	Ddim	B^7, $B♭^7$
38	A♭ minor: V	E♭	A♭ minor: V	E♭	A♭ major: V^7	$E♭^7$	A♭ minor: V	E♭	B♭ major: IV	E♭	E♭
39	A♭ minor: I	**A♭m**	A♭ minor: I	**A♭m**		$\mathbf{E♭}^7$	A♭ minor: I	**A♭m**	C minor: III	**E♭aug**	$D♭^7$ (9)
40	E♭ major: I	E♭	E♭ major: III	Gm		$E♭^7$	E♭ major: III	Gm	B♭ major: VI	Gm	E♭, Gm
41	A♭ major: V	E♭	E♭ major: I	E♭		$E♭^7$	E♭ major: I	E♭	A♭ major: III	Cm	Cm^7, E♭6
42	A♭ major: I	A♭	E♭ major: II	Fm	A♭ major: VI	Fm	E♭ major: II	Fm	A♭ major: I	A♭	A♭6
43	C minor: IV	Fm		Fm		Fm		Fm	A♭ major: VI	Fm	A♭6
44	G minor: II	Adim	G minor: II	Adim	B♭ minor: V^7	F^7	G minor: II	Adim	B♭ minor: VII	Adim	Adim
45	G minor: VII	G♭dim		Adim		F^7		Adim	E minor: VII	E♭dim	Adim
46	A♭ minor: III	Baug	C minor: III	E♭aug	A♭ major: V^7	$E♭^7$	C minor: III	E♭aug	A♭ minor: III	Baug	E♭ (?E♭aug)
47	E♭ major: I	E♭	B♭ major: II	**Cm**		$E♭^7$	B♭ major: II	**Cm**	A♭ minor: V	E♭	E♭ (?E♭aug)
48	B♭ major: V	F	B♭ major: V	F	B♭ minor: V^7	F^7	B♭ major: V	F	B♭ major: V	F	F^7
49		F		F		F^7		F	B♭ minor: V	F	F^7
50	E♭ major: I	E♭	B♭ major: IV	E♭	A♭ major: V^7	$E♭^7$	A♭ major: V	E♭	B♭ major: IV	E♭	E♭
51		E♭		E♭	G minor: VI	E♭		E♭		E♭	E♭
52	E minor: V	B	G minor: II	Adim	E minor: V^7	B^7	G minor: II	Adim	E minor: V	B	B^7
54	E♭ major: II	**Fm**	E♭ major: II	**Fm**	E♭ major: V^7	$\mathbf{B♭}^7$	E♭ major: II	**Fm**	A♭ major: VI	**Fm**	E♭6
55	E♭ major: I	**E♭**	G minor: II	Adim	G minor: V^7	$\mathbf{D}^7$	G minor: II	Adim	G minor: II	Adim	A♭, Adim
56	E♭ major: II	Fm	E♭ major: II	Fm	E♭ major: V^7	$\mathbf{B♭}^7$	E♭ major: II	Fm	E♭ major: II	Fm	A♭
57	E♭ major: V	B♭	E♭ major: V	B♭		$B♭^7$	E♭ major: V	B♭	E♭ major: V	B♭	$B♭^7$
58	E♭ major: I	E♭	E♭ major: I	E♭	A♭ major: V^7	$E♭^7$	E♭ major: I	E♭	A♭ major: V	E♭	E♭6
59	A♭ minor: II	B♭dim	B major: III	**E♭m**	B major: V^7	$G♭^7$	B major: III	**E♭m**	B major: III	**E♭m**	$G♭^7$

Continued...

Bar	**Haydn**		**Jazz**		**Jazz7**		**JazzFun**		**JazzUnigram**		**Manual**
60	E♭ major: II	Fm	E♭ major: II	Fm	E♭ major: V^7	$B♭^7$	E♭ major: II	Fm	E♭ major: II	Fm	Fm^7, $B♭^7$
61	E♭ major: V	B♭	E♭ major: V	B♭		$B♭^7$	E♭ major: V	B♭	E♭ major: V	B♭	B♭
62	E♭ major: I	E♭	E♭ major: I	E♭	A♭ major: V^7	$E♭^7$	E♭ major: I	E♭	B♭ major: IV	E♭	$E♭M^7$
63	F minor: V	C	F minor: V	C	F minor: V^7	C^7	F minor: V	C	G major: II	Am	Am^7 (♭11)
64	C minor: IV	**Fm**	D♭ major: V	A♭	D♭ major: V^7	$A♭^7$	D♭ major: V	A♭	C minor: VI	A♭	$A♭^7$
65		**Fm**	G♭ major: V	D♭	G♭ major: V^7	$D♭^7$	G♭ major: II	**A♭m**	G♭ major: V	D♭	$D♭^7$ (9)
66	B♭ major: IV	E♭	B♭ major: IV	E♭	A♭ major: V^7	$E♭^7$	B♭ major: IV	E♭	B♭ major: IV	E♭	E♭
67	G minor: IV	Cm	G minor: IV	Cm	G minor: V^7	$\mathbf{D}^7$	G minor: IV	Cm	B♭ major: II	Cm	Cm6
68	G♭ major: IV	B	E major: V	B	E major: V^7	B^7	E major: V	B	E minor: V	B	B^7
69	E♭ minor: II	Fdim	E♭ major: V	B♭	E♭ minor: V^7	$B♭^7$	E♭ major: V	B♭	E♭ minor: VII	Ddim	B^7, $B♭^7$
		53/67		*51/67*		*48/67*		*53/67*		*58/67*	

Discussion

The table of 4.6.3 is intended as an insight into the comparative behaviours of the models, rather than as a quantitative evaluation. I have only annotated the validity of the chord labels assigned by the model, not the more informative, but harder, harmonic analysis. A greater insight into the models' behaviour is gained by examining the harmonic labels – key and functional chord – that they assign.

The first thing to note is that all the models perform poorly in the opening section, which is in fact an intro. This is simply because it is an especially difficult passage, with no great harmonic coherence. Once the tune begins, at time unit 10, we can expect more of the models.

Most of the models are reasonably successful at identifying the most common instance of an extended cadence – IIm^7 V^7 I – which occurs quite often in this sequence. This suggests that my conclusion above hat most of the models have learned these chord functions may be correct.

The models may be said to have identified the overall key of the piece, E♭, on the basis that this key is heavily represented in the key labels. There is even some sign that HAYDN, JAZZ7 and, interestingly, JAZZUNIGRAM pick up on the modulation to the subdominant, A♭, for the B-section beginning at 42.

JAZZ7 massively overuses its V7 chord, almost every chord being treated as a dominant seventh. In some cases this will be due to the flat seventh appearing as a blue note. This odd and discouraging behaviour could be predicted from the model parameters: almost every key change will be followed by a V7 chord (probability 0.96); and having used a V7 it remains on the same chord with probability 0.75. It is clear that the model has learnt during training to overuse this function. One possible reason for this is the way I chose to incorporate the tetrad chord into the model (a detail omitted from the paper). I treated the seventh note as if it were a fifth, hoping that this would have the effect of grouping it together with the chord notes without having to introduce another note category to the distribution, resulting in greater sparsity. A chord with the flat seventh note has much greater functional ambiguity in jazz than in Haydn's music, but is often an important clue in identifying a chord's function. A model intended to handle jazz music should be given greater flexibility to learn the relationship between this note and the chord's function.

The evaluation of chord recognition performance shows that the unigram model does a better job of selecting an analysis that results in correct chord labels than any other model. The model is not constrained to produce an analysis in which key and chord transitions are consistent with those in the training data, but can freely choose any analysis for each chord which predicts the notes observed. However, even ignoring the harmonic analysis and looking only at the chord label, as we are here, one would hope that the learned transition probabilities would help boost performance. It is worrying that the benefits of this information, which can be seen at least moderately in the harmonic labels, do not show up in the chord labels.

Perhaps more surprising, though, is the scale of the difference. JAZZUNIGRAM performs vastly better at choosing chord labels, making it clear that the other models make a substantial sacrifice by choosing in interpretation with a more coherent key and chord analysis. Nevertheless, a glance at the actual analyses the models output shows that the contextual information is helping

the models to produce a coherent analysis, which is what is of most relevance to our adaptation of the models to the supertagging task. The evaluation at the level of chord labels is misleading here. In order to get a better idea of how the performance of the unigram model compares to that of the others we need to find a way of evaluating the quality of the harmonic labels. This remains an open problem.

4.7 Conclusions of Experiments on the R&S Models

I began this chapter by asking two questions:

- Can the R&S model can be applied to jazz standard MIDI files and successfully learn the same sort of information it is able to learn from classical music? and
- Are the model's analyses sufficiently accurate that they could serve as a good starting point for a MIDI supertagging model?

The answer to the first of these questions appears from the above experiments to be that it can indeed. I discussed various reasons why the model might have trouble modeling the data of jazz standard MIDI files in the way it did Haydn sonata MIDI files. As predicted, the patterns learned by the model when applied to jazz data are weaker than those seen in the Haydn models. But nevertheless, many of the same patterns can be seen in some form. A particularly notable result is that the simple rhythmic component proposed for the R&S model still seems useful on the jazz data: although the distinctions are weaker, there is a substantial difference between the distributions learned for the different rhythmic positions within a bar, and the differences follow the same pattern as in the Haydn models.

The second question is harder to answer from the experiments reported here. We have seen that certain forms of the model, though not all, showed signs in their learned distributions of having captured some notion of dominant and subdominant function. This is mildly encouraging for the proposed MIDI supertagging models.

Beyond this, it is hard to draw any strong conclusions in answer to the second question. It seems that a model like the R&S model should be able to handle the jazz data at least moderately well. The only way to get a better idea of how well the proposed models will work now is to try implementing them and to evaluate their performance as supertagging models.

4.8 Supertagging Model

The next step is to think about how the model could be adapted to be used as a MIDI supertagger, taking into consideration the experiments and observations above.

The basic form of a supertagging model must be as follows. The states of the model are a combination of a CCG lexical schema *lex* chosen from those defined in the jazz grammar and a root pitch class *root*. Together these represent a CCG

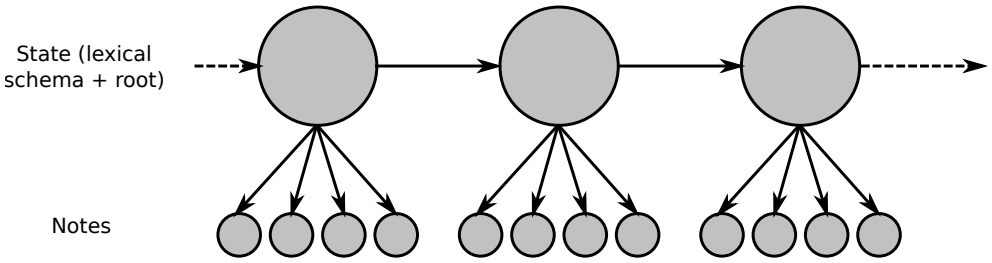


Figure 4.1: The structure of an HMM such as that used in the original R&S model, now adapted to the supertagging task.

lexical category, since the lexical schema, which generalizes over roots, may be instantiated at this root to produce a lexical category.

There are 29 lexical schemata to choose from[2]. Each may be combined with one of 12 roots to constitute a state label, giving 348 states. This means there are 121,104 state transitions. However, if we tie parameters in the same way as the R&S model to make the model ignore absolute pitch, the number of state transitions is reduced to $29 \times 29 \times 12 = 10,092$.

The emission distribution is then defined as $P(O|lex, root)$, where O is the set of notes in a bar. We may define this differently in order to distinguish between the probabilities of generating notes at different times in the bar. The R&S model uses a method of doing this which maintains independent probability distributions for notes that occur on different beats, or off the beat.

4.8.1 Tying the States' Emission Distributions

It is a feature of the grammar that multiple CCG categories may be assigned to the same chord and represent different structural (and semantic) interpretations of the chord. Some chords are ambiguous because they do not express certain clues that would rule out some interpretations: an open fifth may be an under-expressed major or minor chord Some interpretations are never distinguished in the surface form: a subdominant chord always looks the same as a tonic chord; a tritone-substituted dominant chord is always identical to the dominant chord a tritone away from it.

One approach to this latter type of ambiguity is to model the note distributions of each category separately and hope that the model learns similar distributions for these ambiguities. The structure of such a direct adaption of the R&S HMM to the supertagging task is shown in figure 4.1. However, the shortage of data means that with as many distributions to learn as the number of states, we will be left with very little evidence for any particular distribution. What is more, we already have written into the lexicon for parsing chord sequences a generalization of chord types that ought to be useful for this purpose.

[2]There are 44 entries in the lexicon, of which 18 are repetition categories (that is, X/X or X\X categories, with different root substitutions and functions). The tonic function schemata must be included, since they are required to build certain structures, such as a I IV I interpretation. Those with other functions only exist to handle repeated chords, but the supertagging model will run together consecutive bars into a single span where a lexical interpretation spans more than one bar. This leaves only 29 categories.

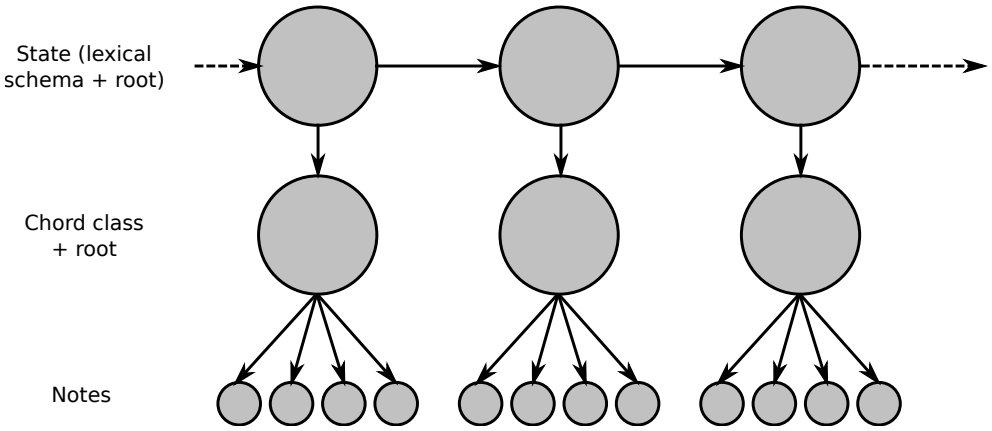


Figure 4.2: A suggested extension to the basic HMM for MIDI supertagging, including an extra layer to represent chord classes.

Each lexical schema in the lexicon is associated with a class of chord types: one of X, Xm, X^7, Xm^7 and $X\circ^7$. In order to use this lexicon for parsing chord sequences, we defined each of these classes as a set of chord types. If a chord falls into the set that defines a chord class, it may be assigned the lexical category formed from any lexical schema that has the chord class on its left-hand side.

The generalization that we would be insane not to exploit here is that any lexical schema associated with X^7, for example, should emit pitch classes with the same, or very similar, probabilities to any other lexical schema associated with X^7 (to use the terminology of a hidden Markov model where the lexical entries are the states). There are two ways that immediately spring out of this to structure a model to exploit this information, described in the following sections.

Model Structure Extension

The first way to exploit the above generalizaion is to add an extra layer into the model between the states and the pitch-class emissions. A state (lexical schema, plus root pitch class), instead of directly emitting the observed notes, emits a label of a chord class (X, X^7, etc), carrying with it the root pitch class. This intermediate state then emits the notes themselves. This means that all states associated with X^7, for instance, given the same root (that is, value of X) emit notes from the same probability distribution. The transition distributions are, of course, parameterized according to the full set of state labels (lexical entries), so that, during decoding, states associated with the same chord class will be distinguished by their transition probabilities. The emission distributions would need to be initialized naively in a similar way to those in the R&S model. For example, X would have a high probability of the root, major third and fifth and a fairly high probability of the major seventh; whilst X^7 would have a high probability of the flattened seventh. The structure of this extended model is shown in figure 4.2.

As an example of how this might behave, let us consider a state label (Dom, G), the lexical schema for unsubstituted dominant function categories, with

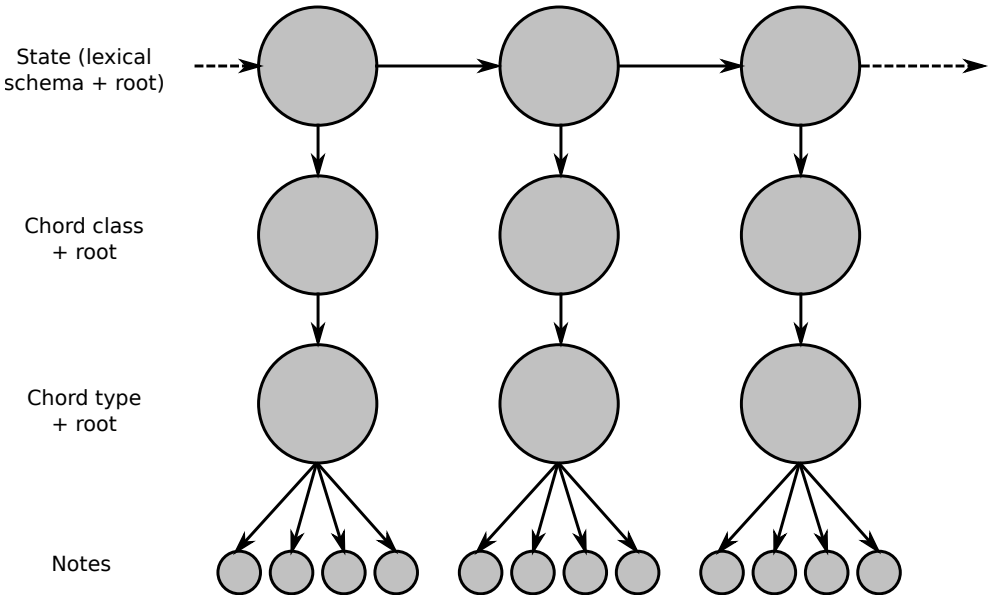


Figure 4.3: A suggested further extension to the MIDI supertagging model, now including an additional layer to represent chord types.

the pitch-class root G. (See figure 2.13 for details of lexical schemata.) This will necessarily generate the chord classes X^7 and Xm^7, since the schema's definition allows it to be used by both major and minor dominant chords. The simplest way to do this would be to define a distribution in which the state may generate either of these chord classes with probability 0.5. Alternatively, the parameters of this distribution could be learned during training. The chord class states generated at the intermediate level will be (X^7, G) and (Xm^7, G). Note that these chord class states could also have been generated by other lexical schemata, such as Dom-tritone. The chord class state X^7 (and likewise Xm^7) then generates the pitch classes of the observed notes according to an emission distribution. This distribution will look something like the emission distribution of the R&S model. I discuss this further in section 4.8.2.

Further Model Structure Extension

The approach described above may be too coarse-grained a generalization of the emission probabilities. Each chord class is defined as a set of chord types. Let us say that one class contains the chord types Xm and Xdim (the diminished triad, often notated Xm,♭5). If the emission distribution is conditioned only on chord class, instances of both of these chord types must be generated by the same distribution. The first chord contains the root, minor third and fifth, the second the root, minor third and flattened fifth. The distribution would learn that both the fifth and flattened fifth may be emitted by this class, but not that the two rarely occur in the same chord.

This leads us to a second approach in which yet another layer is added to the distribution representing the chord types themselves. The states would generate chord classes, as in the previous suggestion, but these would now generate chord

types, with a probability distribution that would be trained as part of the EM procedure. These chord labels would then generate the notes themselves. The structure of this second extension to the model is shown in figure 4.3.

Let us extend the above example to this second approach. As before, the state (Dom, G) generates (X^7, G) and (Xm^7, G) with some probability. The chord class state (X^7, G) then generates a chord state, according to some probability distribution which assigns a non-zero probability to each chord type defined as being contained in that class. This is not conditioned on the root, G, but only the chord class, X^7. Let us say that the chord type $X^{7,\flat 9}$ is generated. The chord state, at the new intermediate level, is now (X^7, $\flat$9, G), or $G^{7,\flat 9}$. It is this state which then generates the observed notes. The chord class state could also have generated the chord state $G\phi^7$ (a half-diminished chord), which would have generated notes according to an independent distribution.

This second suggestion presents two potential problems. The first is data sparsity: many of these chord types will now have few examples in the training data from which to learn their note emission distributions and the chord-class to chord emission distributions. The second is that I am not certain whether the convergence guarantees of EM for HMMs would still hold for training this model, since there is a many-to-many relationship between the chord labels and the chord classes. One major advantage to this second approach is that, if it can be successfully trained, the decoding procedure would produce not only a full harmonic analysis – the aim of our task – but also a set of chord labels produced on the basis of a model which attempts to satisfy constraints on the harmonic analysis, rather than merely constraints on the chord transitions.

4.8.2 Ideas for the Emission Distribution

The R&S model defines an emission distribution that is structured using triads and scales. As a reminder:

$$p(y^k|x,r) = \frac{p(d(y^k,x)|r^k)}{V(d(y^k,x))}$$

where

$$d(y^k,x) = d(y^k,t,m,c) = \begin{cases} 1 & y^k \textit{ is root of chord } t,m,c \\ 2 & y^k \textit{ is 3rd of chord } t,m,c \\ 3 & y^k \textit{ is 5th of chord } t,m,c \\ 4 & y^k \textit{ is in scale } t,m \\ 5 & \textit{otherwise} \end{cases}$$

In section 4.8.1, I defined a model in which notes are emitted from states representing chord classes. Such an emission distribution could be structured like the R&S model's as follows. The d function could be redefined for states made from chord classes c and roots r:

$$d(y^k,x) = d(y^k,c,r) = \begin{cases} 1 & y^k = r \\ 2 & y^k \in \textit{tetrad}(c,r) \\ 3 & y^k \in \textit{scale}(c,r) \\ 4 & y^k \textit{ otherwise} \end{cases}$$

where *tetrad* and *scale* define sets of pitch classes. For example:

$$tetrad(X, r) = \{r + 4, r + 7, r + 11\}$$
$$tetrad(Xm^7, r) = \{r + 3, r + 7, r + 10\}$$

etc.

$$scale(X, r) = scale_M(r)$$

that is, the major scale of r.

$$scale(X^7, r) = scale_M(r - 7)$$

which is the mixolydian mode on r.

$$scale(Xm^7, r) = scale_M(r - 2)$$

which is the dorian mode on r.

$$scale(X\circ^7, r) = \emptyset$$

since we should consider the diminished chord class ambiguous as to its scale.

Alternatively, we could abandon the abstraction of the d funcion altogether and simply learn a distribution over all 12 pitch classes for each state (and rhythm position, if we use that to condition the distributions). This results in just over twice as many parameters to learn for each distribution, but means that we no longer make the assumption previously encoded in the d function about the structure of these parameters: namely that all non-triad notes that are in the chord's scale have the same probability. This assumption may have been reasonable for the original R&S model, but it is not clear that it is appropriate for this model, since the concept of a state's scale is less clearly defined.

4.8.3 Segmentation

The R&S model handles the problem of segmentation by defining a minimal time unit (a bar or half a bar) and running together consecutive units that remain in the same state. This requires certain information to be associated with the MIDI data before analysis begins: the length of the time unit and the time at which the first time unit begins.

Whilst it would be desirable not to require these hand-specified parameters for each MIDI sequence that is analyzed, accepting MIDI input without them would require a sophisticated metrical model to identify the metrical structure. This could be done as a pre-processing step before harmonic analysis begins, or optimised simultaneously with the harmonic structure, since the two structures are not orthogonal. In any case, for now it seems reasonable to continue to require this information to be given by hand, since modeling metrical structure remains a difficult open problem (Temperley (2007) presents an overview of some of the probabilistic approaches explored to date).

A useful feature of the grammar is that the only reason why an analysis would ever require the same lexical category (that is the same lexical schema paired with the same root) to occur more than once consecutively is because a chord is repeated. This happens occasionally in chord sequences merely due to notational issues (a new line or section begins while the same chord is sustained).

The purpose of a model of harmonic segmentation, however, is precisely to ensure that this never occurs: a new category should be used only when there is a harmonic change.

There is, therefore, no circumstance in which the same combination of lexical schema and root (that is the same state in our models) should occur twice in a row. As a result, we may take the same approach to segmentation taken in the R&S model. Two consecutive time units with the same state are treated as having a single category spanning them both. The continuation of a chord is thereby modelled as a self-transition in the state sequence.

4.8.4 Evaluating the Models

It will be useful to be able to evaluate the performance of the supertagger in suggesting categories for the parser before putting it to use in combination with the parser and evaluating the system as a whole. This will allow me to compare the model structures suggested above (and any others that may follow from experiments and discussion). I could carry out this evaluation by annotating a small set of MIDI files with gold standard grammatical analyses and measuring the supertagger's category prediction accuracy against this.

I used two metrics to do the same with the chord supertagger. First, I measured the proportion of gold standard categories found in the highest-probability n categories predicted by the supertagger. The supertagger can be considered useable if a high accuracy (something of the order of 95%) can be acheived with n set to 4 or 5, since, in practical use, the supertagger will be allowed to suggest several categories to the parser. Second, I measured the cross-entropy of the supertagger's distribution over categories with the gold standard categories. Essentially, this is simply a measure of the average probability with which the supertagger would suggest each correct category, if the adaptive supertagging process gets as far as finding the gold standard category. The number produced by this metric, being an entropy, is somewhat difficult to interpret, but does provide a good way of comparing competing models. It is based on the intuition that a model which assigns a higher probability to the gold standard categories will give the parser a better chance of finding a good parse, even if it does not manage to use all of the gold standard categories.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Many practical tasks that involve the processing of music require an prior analysis of the structures underlying its harmony. In order to automate these tasks, we first need an automatic mechanism to perform this harmonic analysis. We have begun by tackling a simpler formulation of this problem which involves producing a harmonic analysis of a chord sequence. However, for many tasks it is desirable to be able to operate directly on the music notes as played by an instrument such as an electric piano. We have argued that the same sort of harmonic analysis we carried out first on chord sequences is valid when approaching the more difficult task of understanding the harmony underlying a stream of notes.

Our harmonic analysis is in terms of movements of a tonal centre about the tonal space of Christopher Longuet-Higgins. We described a grammar and that, together with standard statistical modeling techniques from NLP, allows us to perform this sort of harmonic analysis. It is formulated in such a way that it can operate directly on chord sequences, treating each chord as if it were a word in a natural language parsing task. In our experiments, we show that such a grammatical model, with these basic modeling techniques, is able to outperform a closely related shallower statistical model, namely a hidden Markov model, which does not use a grammar.

We also went on to show how the shallow model could be combined with the grammar-based model to produce a robust method for automatic harmonic analysis of chord sequences. As we mentioned above, though, the form of the analysis itself and certain components of the statistical modeling (namely the parts that direct the combination of grammatical categories into a parse tree) are not restricted to such a use on chord sequences. By replacing the part of the system that associates grammatical categories with words, we can also perform the grammatical analysis on streams of notes, allowing us to attempt the more difficult of the two tasks mentioned above.

We have introduced a related model, that described by Raphael & Stoddard (2004). Although the harmonic analysis that this model is designed to produce is

of a different sort to ours, we believe that their approach contains certain insights that could form the basis for the note-processing component of our system. We have therefore presented an analysis of this related model and performed some experiments intended to answer several questions about how well the model could be adapted to our target domain and to our form of harmonic analysis. We have presented some ideas for the structure that our model might have, but have not yet tried implementing this model.

5.2 Future Plans

The next step is to experiment with a supertagger component such as that described above – to try training models with the suggested structures and see whether they can be made to perform sufficiently well in the place of the chord supertagging component that one of them could take its place in a note stream parsing system. If we are successful in this, we must consider how to evaluate the system. The supertagging component itself could be evaluated on the accuracy of the grammatical categories it assigns to the note data, though this would require the annotation of a gold standard for some part of our MIDI file corpus.

A MIDI supertagger could be combined directly with the parsing models that we have trained on the annotated corpus of chord sequences. This combination is not entirely natural, since the statistics of the parsing model are trained on the chord corpus, for which a slightly different set of grammatical categories was appropriate[1]. An alternative approach would be to train the two models – the supertagging model and the parsing model – together. We use Baum-Welch to train the supertagger and could use the closely related *inside-outside algorithm* the train the parsing model. These two algorithms can be combined using a method such as dual decomposition or belief propagation to optimize the two models simultaneously, as in Auli & Lopez (2011a). However, we do not have time to do this as part of the current project, so will satisfy ourselves with reusing the statistics from the annotated chord corpus for the MIDI parsing task.

We plan to evaluate the full system by applying it to a specific task: a form of *music information retreival* (MIR). MIR is the musical equivalent of *information retreival*, as the term is used in NLP. It encompasses all forms of musical searching and database querying. The task we intend to tackle is that of identifying a song given MIDI data from one performance of the song, making use of a corpus of MIDI files including at least one example of the same song. Different performances may vary in instrumentation, style of performance, key and so on. They may not even use the same chord sequence, but vary it, for example, by use of substitutions. Our hypothesis is that a system that does this by comparing harmonic analyses of those songs in the corpus to that of the query song will perform better than one that looks only for similarity in the note streams. We will also be able to compare the performance of our system against one that uses the shallower harmonic analyses produced by the Raphael and Stoddard system in the same way.

[1] Recall that the category set for the MIDI parsing task is slightly smaller than that for the chord parsing task.

5.3 Timeline

We hope to complete all of the above the next six to eight months. The table below sets out a rough timeline for this work and the subsequent thesis-writing phase, ending with thesis submission a little over three years after I began the PhD project.

Dates	*PhD months*	*Work*
Oct 2011	24	Submission of present report.
12 Oct 2011	24	Delivery of second-year review presentation.
Oct–Dec 2011	24–26	Set up MIR task and evaluation framework, using R&S model initially.
Jan 2012	27	Establish a baseline MIR model using a note stream comparison metric directly. Evaluate this against R&S model.
Feb–Mar 2012	28–29	Try applying HMM-type models to MIDI supertagging task. Annotate a small sample of MIDI corpus and evaluate supertagger accuracy.
Apr 2012	30	Use MIDI supertagger with parsing models and evaluate on MIR task.
Apr 2012	30	Begin writing up.
May 2012	31	Draft dissertation defence.
Nov 2012	37	Submit thesis.

Bibliography

Auli, M., & Lopez, A. (2011a). A comparison of loopy belief propagation and dual decomposition for integrated ccg supertagging and parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, (pp. 470–480). Stroudsburg, PA, USA: Association for Computational Linguistics.

Auli, M., & Lopez, A. (2011b). Training a log-linear parser with loss functions via softmax-margin. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (pp. 333–343). Edinburgh: ACL.

Bernstein, L. (1976). *The Unanswered Question*. Cambridge, MA: Harvard University Press.

Bod, R. (2002). Memory-based models of melodic analysis: Challenging the gestalt principles. *Journal of New Music Research*, *31*, 27–37.

Clark, S., & Curran, J. R. (2007). Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, *33*, 493–552.

Cohn, R. (1997). Neo-riemannian operations, parsimonious trichords, and their *tonnetz* representation. *Journal of Music Theory*, *41*, 1–66.

Cooke, D. (1959). *The Language of Music*. Oxford: Oxford University Press.

Cooper, G., & Meyer, L. B. (1963). *The Rhythmic Structure of Music*. University of Chicago Press.

Daniélou, A. (1968). *The Rāga-s of Northern Indian Music*. London: Barrie & Rockliff, The Cresset Press.

Euler, L. (1739). *Tentamen novae theoriae musicae ex certissismis harmoniae principiis dilucide expositae*. Saint Petersberg Academy. Tonnetz p.147.

Hal Leonard Corp. (2006). *The Real Book, Sixth Edition*. Hal Leonard Europe.

Helmholtz, H. (1862). *Die Lehre von dem Tonempfindungen*. Braunschweig: Vieweg. Trans. Alexander Ellis (1875, with added notes and appendices) as *On the Sensations of Tone*.

Hockenmaier, J., & Steedman, M. (2002). Generative models for statistical parsing with Combinatory Categorial Grammar. In *Proceedings of the 40th Meeting of the ACL*, (pp. 335–342). Philadelphia, PA.

Honingh, A., & Bod, R. (2005). Convexity and well-formedness of musical objects. *Journal of New Music Research*, *34*, 293–303.

Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Music*. Cambridge, MA: MIT Press.

James, W. (1884). What is an emotion? *Mind*, *9*, 188–205.

Jeans, J. (1937). *Science and Music*. Cambridge: Cambridge University Press.

Keiler, A. (1981). Two views of musical semiotics. In W. Steiner (Ed.) *The Sign in Music and Literature*, (pp. 138–168). Austin TX: University of Texas Press.

Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press.

Lerdahl, F., & Jackendoff, R. (1996). *A Generative Theory of Tonal Music*. The MIT Press.

Liberman, M. (1975). *The Intonational System of English*. Ph.D. thesis, MIT. Published by Garland Press, New York, 1979.

Lindblom, B., & Sundberg, J. (1969). Towards a generative theory of melody. *Swedish Journal of Musicology*, *10*(4), 53–86.

Longuet-Higgins, H., & Steedman, M. (1971). On interpreting bach. *Machine Intelligence*, *6*, 221–241.

Longuet-Higgins, H. C. (1962a). Letter to a musical friend. *The Music Review*, *23*, 244–248.

Longuet-Higgins, H. C. (1962b). Second letter to a musical friend. *The Music Review*, *23*, 271–280.

Longuet-Higgins, H. C. (1979). Perception of melodies. *Nature*, *263*, 646–653.

Margulis, E. H. (2005). A model of melodic expectation. *Music Perception*, *22*, 663–714.

Mathews, M. V., & Pierce, J. R. (1989). The bohlen-pierce scale. (pp. 165–173). Cambridge, MA: MIT Press.

Meyer, L. (1956). *Emotion and Meaning in Music*. Chicago, IL: University of Chicago Press.

Narmour, E. (1977). *Beyond Schenkerism*. Chicago, IL: University of Chicago Press.

Plomp, R., & Levelt, W. (1965). Tonal consonance and critical bandwidth. *Journal of the Acoustical Society of America*, *38*, 518–560.

Raphael, C., & Stoddard, J. (2004). Functional harmonic analysis using probabilistic models. *Computer Music Journal*, *28*(3), 45–52.

Riemann, H. (1914). Ideen zu einer Lehre von den Tonvorstellungen. *Jahrbuch der Bibliothek Peters*, *21*, 1–26.

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

Sartre, J.-P. (1947). Une idée fondamontale de la phénoménologie de Husserl: l'Intentionnalité. In J.-P. Sartre (Ed.) *Situations*, vol. I, (pp. 31–35). Paris: Gallimard.

Savitch, W. (1989). A formal model for context-free languages augmented with reduplication. *Computational Linguistics*, *15*, 250–261.

Srinivas, B., & Joshi, A. (1994). Disambiguation of super parts of speech (or supertags): Almost parsing. In *Proceedings of the International Conference on Computational Linguistics*. Kyoto: ACL.

Steedman, M. (2000). *The Syntactic Process*. Cambridge, MA, USA: MIT Press.

Temperley, D. (2007). *Music and Probability*. Cambridge, MA: MIT Press.

Tymoczko, D. (2006). The geometry of musical chords. *Science*, *313*, 72–74.

Tymoczko, D. (2011). *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford: Oxford University Press.

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, *12*, 2–49.

Appendix A

R&S Model Parameters

A.1 Haydn Parameters

There follow the parameters of the distributions that resulted from training the HAYDN model (see section 4.5.1).

A.1.1 Emission Distribution

Beat category: 0 (1st beat)

D = 0	0.36288	(chord root)
D = 1	0.30916	(chord 3rd)
D = 2	0.22071	(chord 5th)
D = 3	0.09815	(other scale note)
D = 4	0.00910	(non-scale note)

Beat category: 1 (3rd beat)

D = 0	0.33418	(chord root)
D = 1	0.26080	(chord 3rd)
D = 2	0.19181	(chord 5th)
D = 3	0.19778	(other scale note)
D = 4	0.01542	(non-scale note)

Beat category: 2 (2nd or 4th beat)

D = 0	0.19641	(chord root)
D = 1	0.20286	(chord 3rd)
D = 2	0.16686	(chord 5th)
D = 3	0.39815	(other scale note)
D = 4	0.03573	(non-scale note)

Beat category: 3 (off beat)

D = 0	0.16615	(chord root)
D = 1	0.19531	(chord 3rd)
D = 2	0.21370	(chord 5th)
D = 3	0.38990	(other scale note)
D = 4	0.03493	(non-scale note)

A.1.2 Key Transition Distribution

Previous mode: major

Key	Mode	
I	major	0.90289
VI	minor	0.03788
IV	major	0.02226
V	major	0.01888
II	minor	0.00782
III	minor	0.00493
bIII	major	0.00212
bVI	major	0.00188
IV	minor	0.00134
bV	major	0.00000
V	minor	0.00000
VII	minor	0.00000
bIII	minor	0.00000
II	major	0.00000
I	minor	0.00000
bVII	major	0.00000
bII	minor	0.00000
VI	major	0.00000
bVII	minor	0.00000
bII	major	0.00000
bVI	minor	0.00000
III	major	0.00000
VII	major	0.00000
bV	minor	0.00000

Previous mode: minor

Key	Mode	
I	minor	0.60519
bVII	major	0.11517
IV	minor	0.09082
bVI	major	0.06541
V	minor	0.04536
VII	major	0.02285
bII	minor	0.01882
V	major	0.01565
IV	major	0.01429
I	major	0.00630
VI	major	0.00014
bIII	major	0.00000
bVII	minor	0.00000
III	minor	0.00000
VI	minor	0.00000
bV	major	0.00000
II	minor	0.00000
bII	major	0.00000
VII	minor	0.00000
III	major	0.00000
bVI	minor	0.00000
II	major	0.00000
bIII	minor	0.00000
bV	minor	0.00000

A.1.3 Chord Transition Distribution

Previous chord: I

I	0.62011
VII	0.12544
V	0.10546
IV	0.07790
II	0.04895
VI	0.02215
III	0.00000

Previous chord: II

V	0.46517
I	0.33481
VII	0.10516
II	0.09486
IV	0.00000
VI	0.00000
III	0.00000

Previous chord: III

III	0.83150
I	0.16850
V	0.00000
II	0.00000
IV	0.00000
VI	0.00000
VII	0.00000

Previous chord: IV

I	0.75704
IV	0.16027
VII	0.07765
II	0.00504
V	0.00000
VI	0.00000
III	0.00000

Previous chord: V

I	0.75206
V	0.11487
II	0.07120
VI	0.03663
VII	0.02525
III	0.00000
IV	0.00000

Previous chord: VI

II	0.56294
IV	0.43706
V	0.00000
VI	0.00000
I	0.00000
III	0.00000
VII	0.00000

Previous chord: VII

I	0.89949
II	0.05872
VII	0.04179
IV	0.00000
V	0.00000
VI	0.00000
III	0.00000

A.1.4 Key Change Chord Distribution

I	0.31154
V	0.23810
II	0.15392
VII	0.12090
III	0.11074
IV	0.06480
VI	0.00000

A.2 Haydn7 Parameters

There follow the parameters of the distributions that resulted from training the HAYDN7 model (see section 4.5.2).

A.2.1 Emission Distribution

Beat category: 0 (1st beat)

D = 0	0.31118	(chord root)
D = 1	0.25836	(chord 3rd)
D = 2	0.31701	(chord 5th)
D = 3	0.10725	(other scale note)
D = 4	0.00620	(non-scale note)

Beat category: 1 (3rd beat)

D = 0	0.28398	(chord root)
D = 1	0.23825	(chord 3rd)
D = 2	0.31291	(chord 5th)
D = 3	0.15179	(other scale note)
D = 4	0.01307	(non-scale note)

Beat category: 2 (2nd or 4th beat)

D = 0	0.18115	(chord root)
D = 1	0.21385	(chord 3rd)
D = 2	0.25374	(chord 5th)
D = 3	0.31615	(other scale note)
D = 4	0.03512	(non-scale note)

Beat category: 3 (off beat)

D = 0	0.17280	(chord root)
D = 1	0.19209	(chord 3rd)
D = 2	0.26633	(chord 5th)
D = 3	0.33394	(other scale note)
D = 4	0.03484	(non-scale note)

A.2.2 Key Transition Distribution

Previous mode: major

Key	Mode	
I	major	0.86087
VI	minor	0.04679
IV	major	0.04430
V	major	0.03445
II	minor	0.00452
III	minor	0.00410
bVI	major	0.00195
IV	minor	0.00192
bIII	major	0.00109
bV	major	0.00000
bIII	minor	0.00000
bVII	minor	0.00000
bVII	major	0.00000
II	major	0.00000
I	minor	0.00000
V	minor	0.00000
VI	major	0.00000
VII	minor	0.00000
bII	major	0.00000
bV	minor	0.00000
bII	minor	0.00000
bVI	minor	0.00000
VII	major	0.00000
III	major	0.00000

Previous mode: minor

Key	Mode	
I	minor	0.57960
bVII	major	0.13833
IV	minor	0.09388
bVI	major	0.06118
V	minor	0.04034
VII	major	0.02240
bII	minor	0.01953
IV	major	0.01234
V	major	0.01076
bV	major	0.00515
VII	minor	0.00503
VI	major	0.00448
I	major	0.00394
bIII	major	0.00305
bVII	minor	0.00000
bIII	minor	0.00000
VI	minor	0.00000
III	minor	0.00000
II	minor	0.00000
bII	major	0.00000
bVI	minor	0.00000
II	major	0.00000
bV	minor	0.00000
III	major	0.00000

A.2.3 Chord Transition Distribution

Previous chord: I

V7	0.49156
I	0.48031
VI	0.02813
II	0.00000
IV	0.00000
III	0.00000
V	0.00000
VII	0.00000

Previous chord: II

V7	1.00000
I	0.00000
VII	0.00000
V	0.00000
II	0.00000
III	0.00000
VI	0.00000
IV	0.00000

Previous chord: III

VI	0.24924
V7	0.14022
I	0.11769
IV	0.10259
II	0.09971
III	0.09855
VII	0.09667
V	0.09534

Previous chord: IV

IV	0.93422
I	0.06578
V7	0.00000
VI	0.00000
II	0.00000
III	0.00000
VII	0.00000
V	0.00000

Previous chord: V

II	0.15344
V7	0.13449
I	0.13001
VI	0.12808
IV	0.11405
III	0.11376
VII	0.11339
V	0.11278

Previous chord: VI

II	0.40396
VI	0.33302
V7	0.20420
IV	0.05882
I	0.00000
III	0.00000
V	0.00000
VII	0.00000

Previous chord: VII

II	0.14990
V7	0.13703
VII	0.12910
I	0.11992
V	0.11888
III	0.11633
VI	0.11538
IV	0.11345

Previous chord: V7

I	0.69149
V7	0.27584
VI	0.01796
II	0.01472
IV	0.00000
V	0.00000
III	0.00000
VII	0.00000

A.2.4 Key Change Chord Distribution

V7	0.66285
I	0.31445
IV	0.01791
VI	0.00337
II	0.00142
III	0.00000
V	0.00000
VII	0.00000

A.3 Jazz Parameters

There follow the parameters of the distributions that resulted from training the Jazz model (see section 4.5.3).

A.3.1 Emission Distribution

Beat category: 0 (1st beat)

D = 0	0.31311	(chord root)
D = 1	0.19380	(chord 3rd)
D = 2	0.20926	(chord 5th)
D = 3	0.25480	(other scale note)
D = 4	0.02903	(non-scale note)

Beat category: 1 (3rd beat)

D = 0	0.28452	(chord root)
D = 1	0.16902	(chord 3rd)
D = 2	0.32592	(chord 5th)
D = 3	0.17221	(other scale note)
D = 4	0.04832	(non-scale note)

Beat category: 2 (2nd or 4th beat)

D = 0	0.20450	(chord root)
D = 1	0.21512	(chord 3rd)
D = 2	0.19698	(chord 5th)
D = 3	0.31851	(other scale note)
D = 4	0.06489	(non-scale note)

Beat category: 3 (off beat)

D = 0	0.17350	(chord root)
D = 1	0.18209	(chord 3rd)
D = 2	0.16609	(chord 5th)
D = 3	0.38402	(other scale note)
D = 4	0.09430	(non-scale note)

A.3.2 Key Transition Distribution

Previous mode: major

Key	Mode	
I	major	0.79638
III	minor	0.05193
bVII	major	0.03815
VI	minor	0.03044
IV	major	0.01865
II	minor	0.01714
III	major	0.01509
VII	minor	0.00973
V	major	0.00814
II	major	0.00567
VII	major	0.00530
V	minor	0.00166
bVI	major	0.00093
bII	major	0.00077
I	minor	0.00000
bV	minor	0.00000
bIII	major	0.00000
IV	minor	0.00000
VI	major	0.00000
bVI	minor	0.00000
bV	major	0.00000
bVII	minor	0.00000
bII	minor	0.00000
bIII	minor	0.00000

Previous mode: minor

Key	Mode	
I	minor	0.43470
bVII	major	0.18771
bVI	major	0.16887
IV	minor	0.07069
bIII	major	0.05819
bII	major	0.02680
V	minor	0.01699
bVII	minor	0.01290
I	major	0.00950
V	major	0.00623
bV	major	0.00401
VI	major	0.00241
VI	minor	0.00073
IV	major	0.00026
II	minor	0.00000
VII	major	0.00000
III	minor	0.00000
II	major	0.00000
bIII	minor	0.00000
bVI	minor	0.00000
bV	minor	0.00000
bII	minor	0.00000
III	major	0.00000
VII	minor	0.00000

A.3.3 Chord Transition Distribution

Previous chord: I

I	0.67497
II	0.15471
IV	0.09925
VII	0.06820
III	0.00271
VI	0.00016
V	0.00000

Previous chord: II

V	0.60142
II	0.30060
III	0.08519
IV	0.01277
VII	0.00001
VI	0.00001
I	0.00000

Previous chord: III

I	0.42999
III	0.38528
IV	0.11913
V	0.05809
VI	0.00748
VII	0.00004
II	0.00000

Previous chord: IV

VII	0.45148
IV	0.37075
III	0.11032
VI	0.05172
V	0.01571
II	0.00001
I	0.00000

Previous chord: V

I	0.67940
V	0.31406
IV	0.00412
III	0.00235
II	0.00004
VI	0.00002
VII	0.00001

Previous chord: VI

VI	0.84814
I	0.07380
IV	0.06911
V	0.00896
III	0.00000
VII	0.00000
II	0.00000

Previous chord: VII

VII	0.42720
VI	0.32106
III	0.17603
V	0.04105
IV	0.02465
I	0.01000
II	0.00002

A.3.4 Key Change Chord Distribution

II	0.45648
V	0.15977
VII	0.15874
IV	0.11401
III	0.08431
VI	0.02565
I	0.00104

A.4 Jazz7 Parameters

There follow the parameters of the distributions that resulted from training the Jazz7 model (see section 4.5.4).

A.4.1 Emission Distribution

Beat category: 0 (1st beat)

D = 0	0.24477	(chord root)
D = 1	0.17444	(chord 3rd)
D = 2	0.34011	(chord 5th)
D = 3	0.21422	(other scale note)
D = 4	0.02646	(non-scale note)

Beat category: 1 (3rd beat)

D = 0	0.21089	(chord root)
D = 1	0.17186	(chord 3rd)
D = 2	0.42392	(chord 5th)
D = 3	0.15126	(other scale note)
D = 4	0.04206	(non-scale note)

Beat category: 2 (2nd or 4th beat)

D = 0	0.18776	(chord root)
D = 1	0.17931	(chord 3rd)
D = 2	0.32419	(chord 5th)
D = 3	0.24131	(other scale note)
D = 4	0.06742	(non-scale note)

Beat category: 3 (off beat)

D = 0	0.16012	(chord root)
D = 1	0.15062	(chord 3rd)
D = 2	0.29032	(chord 5th)
D = 3	0.30290	(other scale note)
D = 4	0.09604	(non-scale note)

A.4.2 Key Transition Distribution

Previous mode: major

Key	Mode	
I	major	0.61598
IV	major	0.09283
V	major	0.06531
VI	minor	0.05730
bVII	major	0.03635
III	minor	0.03542
VII	minor	0.02061
II	major	0.01818
II	minor	0.01571
III	major	0.01374
V	minor	0.00993
bIII	major	0.00648
VII	major	0.00504
VI	major	0.00378
I	minor	0.00210
bVI	major	0.00120
bV	major	0.00004
bII	minor	0.00000
bIII	minor	0.00000
bVI	minor	0.00000
IV	minor	0.00000
bV	minor	0.00000
bII	major	0.00000
bVII	minor	0.00000

Previous mode: minor

Key	Mode	
I	minor	0.25805
bVII	major	0.16788
bVI	major	0.13757
bIII	major	0.11919
IV	minor	0.09637
IV	major	0.06071
bII	major	0.05670
V	minor	0.05100
bV	major	0.01944
VI	minor	0.01650
I	major	0.00819
VI	major	0.00364
V	major	0.00214
VII	major	0.00174
bIII	minor	0.00089
II	minor	0.00000
III	minor	0.00000
bVII	minor	0.00000
bVI	minor	0.00000
II	major	0.00000
bV	minor	0.00000
bII	minor	0.00000
VII	minor	0.00000
III	major	0.00000

A.4.3 Chord Transition Distribution

Previous chord: I

I	0.63405
V7	0.19716
IV	0.12295
II	0.04584
VI	0.00000
III	0.00000
V	0.00000
VII	0.00000

Previous chord: II

V7	0.97962
II	0.02038
III	0.00000
I	0.00000
IV	0.00000
VI	0.00000
V	0.00000
VII	0.00000

Previous chord: III

I	0.74403
III	0.21502
VI	0.04073
IV	0.00022
V7	0.00000
V	0.00000
II	0.00000
VII	0.00000

Previous chord: IV

V7	0.77870
IV	0.20370
III	0.01760
II	0.00000
VI	0.00000
I	0.00000
V	0.00000
VII	0.00000

Previous chord: V

I	0.13478
III	0.13158
V7	0.12984
VI	0.12586
IV	0.12190
V	0.12160
II	0.12042
VII	0.11403

Previous chord: VI

VI	0.78257
IV	0.15660
I	0.06083
V7	0.00000
II	0.00000
III	0.00000
V	0.00000
VII	0.00000

Previous chord: VII

VI	0.12970
V7	0.12778
II	0.12594
IV	0.12586
III	0.12522
I	0.12373
V	0.12119
VII	0.12057

Previous chord: V7

V7	0.74865
I	0.12436
VI	0.06111
IV	0.03386
III	0.03201
II	0.00000
V	0.00000
VII	0.00000

A.4.4 Key Change Chord Distribution

V7	0.95529
VI	0.03330
IV	0.00989
III	0.00151
II	0.00000
I	0.00000
V	0.00000
VII	0.00000

A.5 JazzFun Parameters

There follow the parameters of the distributions that resulted from training the JazzFun model (see section 4.5.5).

A.5.1 Emission Distribution

Beat category: 0 (1st beat)

D = 0	0.31604	(chord root)
D = 1	0.19220	(chord 3rd)
D = 2	0.20723	(chord 5th)
D = 3	0.25518	(other scale note)
D = 4	0.02934	(non-scale note)

Beat category: 1 (3rd beat)

D = 0	0.28360	(chord root)
D = 1	0.17109	(chord 3rd)
D = 2	0.32219	(chord 5th)
D = 3	0.17493	(other scale note)
D = 4	0.04820	(non-scale note)

Beat category: 2 (2nd or 4th beat)

D = 0	0.20553	(chord root)
D = 1	0.21525	(chord 3rd)
D = 2	0.19759	(chord 5th)
D = 3	0.31665	(other scale note)
D = 4	0.06497	(non-scale note)

Beat category: 3 (off beat)

D = 0	0.17331	(chord root)
D = 1	0.18279	(chord 3rd)
D = 2	0.16477	(chord 5th)
D = 3	0.38474	(other scale note)
D = 4	0.09439	(non-scale note)

A.5.2 Key Transition Distribution

Previous mode: major

Key	Mode	
I	major	0.79731
III	minor	0.05162
bVII	major	0.03790
VI	minor	0.03001
IV	major	0.01910
II	minor	0.01622
III	major	0.01497
VII	minor	0.01040
V	major	0.00819
II	major	0.00554
VII	major	0.00532
V	minor	0.00177
bVI	major	0.00090
bII	major	0.00074
bV	minor	0.00000
bIII	major	0.00000
I	minor	0.00000
VI	major	0.00000
IV	minor	0.00000
bV	major	0.00000
bVII	minor	0.00000
bVI	minor	0.00000
bII	minor	0.00000
bIII	minor	0.00000

Previous mode: minor

Key	Mode	
I	minor	0.43141
bVII	major	0.19113
bVI	major	0.17483
IV	minor	0.07173
bIII	major	0.05249
bII	major	0.02692
V	minor	0.01521
bVII	minor	0.01338
I	major	0.01008
V	major	0.00606
bV	major	0.00399
VI	major	0.00245
VI	minor	0.00030
IV	major	0.00000
II	minor	0.00000
VII	major	0.00000
III	minor	0.00000
II	major	0.00000
bIII	minor	0.00000
bVI	minor	0.00000
bV	minor	0.00000
bII	minor	0.00000
III	major	0.00000
VII	minor	0.00000

A.5.3 Chord Transition Distribution

Previous chord: I

I	0.68272
II	0.15505
IV	0.09154
VII	0.07069
III	0.00001
VI	0.00000
V	0.00000

Previous chord: II

V	0.67958
II	0.30434
IV	0.01608
III	0.00001
VI	0.00000
VII	0.00000
I	0.00000

Previous chord: III

III	0.48290
I	0.24750
IV	0.16758
VI	0.05202
V	0.05000
VII	0.00000
II	0.00000

Previous chord: IV

VII	0.43195
IV	0.37892
III	0.13944
VI	0.04956
V	0.00013
II	0.00000
I	0.00000

Previous chord: V

I	0.70196
V	0.29804
IV	0.00000
II	0.00000
III	0.00000
VI	0.00000
VII	0.00000

Previous chord: VI

VI	0.84255
I	0.07517
IV	0.07336
V	0.00891
VII	0.00000
III	0.00000
II	0.00000

Previous chord: VII

VII	0.42617
VI	0.32627
III	0.15804
V	0.04077
IV	0.02483
I	0.02392
II	0.00000

A.5.4 Key Change Chord Distribution

II	0.45664
VII	0.15982
V	0.15513
IV	0.11066
III	0.09860
VI	0.01914
I	0.00001

A.6 JazzUnigram Parameters

There follow the parameters of the distributions that resulted from training the JazzUnigram model (see section 4.5.6).

Only emission distribution parameters are shown, since the transition distributions are all fixed to uniform distributions.

A.6.1 Emission Distribution

Beat category: 0 (1st beat)

D = 0	0.28104	(chord root)
D = 1	0.20468	(chord 3rd)
D = 2	0.21838	(chord 5th)
D = 3	0.27820	(other scale note)
D = 4	0.01770	(non-scale note)

Beat category: 1 (3rd beat)

D = 0	0.28643	(chord root)
D = 1	0.17305	(chord 3rd)
D = 2	0.33761	(chord 5th)
D = 3	0.16402	(other scale note)
D = 4	0.03888	(non-scale note)

Beat category: 2 (2nd or 4th beat)

D = 0	0.21197	(chord root)
D = 1	0.21626	(chord 3rd)
D = 2	0.19456	(chord 5th)
D = 3	0.32915	(other scale note)
D = 4	0.04806	(non-scale note)

Beat category: 3 (off beat)

D = 0	0.17003	(chord root)
D = 1	0.17894	(chord 3rd)
D = 2	0.17045	(chord 5th)
D = 3	0.39023	(other scale note)
D = 4	0.09036	(non-scale note)