School of Informatics

University of Edinburgh

Harmonic Analysis of Music With Combinatory Categorial Grammar

Thesis Proposal

Mark Wilding

June 15, 2010

Contents

Abstract

This project concerns the application of parsing methods used in natural language processing to the task of harmonic analysis of music. Many musical tasks, such as score transcription, rely on an analysis of the harmonic structure underlying tonal music. We use linguistic grammars and statistical parsing techniques to perform this analysis automatically. The huge ambiguity of interpretation in music means that statistical models must play a critical role in restricting the search space.

1 Research Topic and Motivation

Various aspects of musical analysis exhibit hierarchical structures. For example, harmonic progressions group together the notes of polyphonic music; these groups may be further collected into passages of tension and resolution, which in turn may be grouped at a higher level. Independent hierarchical structures are found in rhythmic patterns. These structures in music are analogous to the structures that form and allow us interpret natural language. Formal grammars are used to model the human understanding of language and also to model a computational process for automatic analysis of the meaning of language. In the current work, we apply techniques and formalisms used in natural language processing to the analogous task of musical interpretation.

It is clear that music shares many properties with language: their primary expressions are streams of sound; these streams, between humans, are generally in some sense communicative; the meaning is sensitive to variations in temporal properties of the stream; and so on. Indeed, it is widely believed that musical and linguistic processing share considerable mental faculties and a body of recent neurological evidence supports this,

though how much and what is shared is still much debated (see Patel (2003) for a detailed discussion).

Whether or not one accepts the hypothesis of shared psychological processing, it is at least clear that the structural commonalities between the two make current theories of formal linguistics a good starting point for an equivalent endeavour in musical analysis.

The idea of performing musical analysis in the same terms as linguistic analysis is one with a long history (for example, Cooke (1959), Winograd (1968), Roads and Wieneke (1979), Baroni et al. (1983)). Bernstein (1976) discussed in detail a correspondence between the two domains and attempted abstractly to apply the formal linguistic techniques of the time to musical analysis. Given this history of discussion at an abstract level and the rapid pace of theoretical developments in computational linguistics, it is surprising that so few have attempted an actual application of current linguistic tools to the corresponding problems in music.

What is more, the overlap in human mental machinery for processing music and language has long been suspected to be substantial. Recent neurological research (Patel (2003), Koelsch et al. (2005)) has supported this idea, particularly in the processing of structural elements.

Much work has gone into the development of purely statistical models, mostly unstructured, history-based models, such as n-grams and Markov models (Ponsford et al. (1999), Conklin and Witten (1995), Allan (2002)). Such models in general fail to account for the syntax-like structures underlying music and consequently cannot capture the structural dependencies that are important for many practical applications.

Our approach is different to that suggested by Bernstein, but is consistent with his fundamental ideas: that analysis of structures and meanings in music should be approached in the same way as analysis of language, only largely without reference to any grounded real-world semantics. It follows that we will have best chance of success if we learn from the successes and failures of computational linguistics and draw on the extensive research in the field by analogy to music.

Harmonic analysis of the sort we describe is crucial to intelligent performance of many musical tasks. The following are some tasks which depend on it.

- Automatic music transcription: writing notes on a score according to notational conventions given live performance data.
- Chord sequence generation: producing chord labels given only the a performed melody or polyphonic note input.
- Chord sequence realization: generating the notes of a polyphonic realization given chord labels.
- Melody generation: producing a new melody in a particular style given an accompanying chord sequence.
- Music similarity metrics (for music information retrieval): evaluating the similarity between two pieces of music.
- High-level structure inference: identifying structural elements such as section boundaries and repeated sections.
- Just intonation realization: generating the pitch of each each note of a given (equally tempered) input as it would be played in just intonation (see section 2).

2 Background: Music Theory

2.1 Tuning Theory

The scales used in Western tonal music are founded on the three lowest distinct intervals in the harmonic series (to be precise, the intervals whose pitch ratios are the first three prime factors). These intervals are known as the octave, the perfect fifth and the major third. The two most used triadic chords, the major and minor triads, are formed from these intervals. A tuning of the major and minor scales derived from these intervals as they appear in the three primary chords of a particular key (that rooted on the key note, that a perfect fifth above it and that a perfect fifth below it) is known as *just intonation*.

Historically, musicians have grappled with the problem that arises from the fact that in general they wish to play pieces in different keys on the same instrument, or even to change the key of a piece in the middle. Due to the incommensurability of the three basic intervals (for example, no combination of successive fifths can produce the same interval as a combination of octaves), an instrument that tunes these three chords justly for a given key will need to be retuned to play in another key.

Many alternative tuning systems were devised, all aiming to strike a compromise between preserving the tuning of the natural harmonic intervals in chords and allowing movement between commonly used keys. They also attempted to reduce the unpleasant effect of certain intervals between notes of

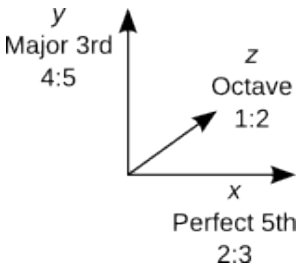

Figure 1: The three dimensions of the tonal space

the justly intoned scale which do not appear in the chords (known as *wolf intervals*).

The tuning system used almost ubiquitously in Western tonal music today is known as *equal temperament*. In this system, the octave interval (a pitch ratio of 2 between two notes) is divided equally into 12 distinct pitches, which are the 12 notes of the chromatic scale. This preserves only the octave interval perfectly, but ensures that the corresponding intervals in any key are identical. This approach was first widely adopted in the 18th century and was strongly resisted for a long time because of the sacrifice of the second and third perfect intervals.

Today our ears are accustomed to the tuning of equal temperament and we no longer raise the objections to its imperfect intervals. However, musical notation still distinguishes many intervals that would be sounded identically in equal temperament, giving rise to the apparent redundancy of distinctions between notes such as $G\sharp$ and $A\flat$. These notes, however, are pitched subtly differently by trained singers and players of instruments without fixed tuning (including many string instruments).

2.2 Tonal Space

Many people have written about the theory of tuning musical instruments (Rameau and Gossett (1971), Helmholtz (1885)) according to the concepts outlined above. Longuet-Higgins (1979) constructed a comprehensive formulation of this commonly accepted theoretical basis for our Western scales and tuning system in terms of a discrete three-dimensional space. The basis vectors of this space are the three perfect intervals (octave, fifth and third). The three dimensions are represented in figure 1. A part of the space (which is, in fact, infinite) is shown in figure 2. In this space, the relationship between justly tempered tones can be simply represented as vectors, including remote intervals between notes in a key and even more remote intervals between notes from different keys. Given the context of a key, shown as regions in the space in figure 3, it is easy to see how notes in the surrounding region should be tuned and the harmonic relation between these notes and those that would be played if the music moved to a new key.

For convenient representation, the space is projected onto two dimensions along the axis of the octave. This has little impact on analysis, since the notes an octave apart tend to be heard as so closely related as to be almost the same notes. It is conventional in analysis to take all intervals modulo the octave, equating for example the perfect eleventh with the perfect fifth. This amounts to the same as this projection onto two dimensions.

It is this space that we adopt as the domain for our analysis of intervals, chords and keys. The harmonic relatedness of two notes can be measured by their distance in this domain, corresponding to their relatedness in the harmonic series, which underlies our perception of music. For example, it is clear from figure 3 that in the key of C major the interval between the D and A found in the scale (an imperfect fifth – three left and one up) is considerably more remote than that between G and C (a perfect fifth – one left).

2.3 Harmonic Analysis

Polyphonic music is commonly analysed as being generated by underlying sequences of chords. These chords may determine what notes are played, but the notes are not restricted to the notes of the chord. A process of substitution allows certain chords, in particular contexts, to be used in place of other chords, potentially with a different root and even chords type. A particularly rich system of substitutions is found in jazz music, but they are also found in classical music in the form of, for example, the German sixth. As a result, the notes played may bear very little relation to the notes of the underlying chord.

A meaningful analysis of the harmony of a piece of tonal music will dictate what chord underlies the harmony at any point in the piece. It must also specify the *harmonic function* of the chord – that is

$G\sharp^-$	$D\sharp^-$	$A\sharp^-$	$E\sharp$	$B\sharp$	$F\sharp\sharp$	$C\sharp$	$G\sharp\sharp^+$	$D\sharp\sharp^+$
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: A region of the tonal space. The note names in the centre mark the notes in a justly tempered system to which the points correspond. Points with + or − are in fact different notes, but borrow their names from the notes of the scale to which they are closest.

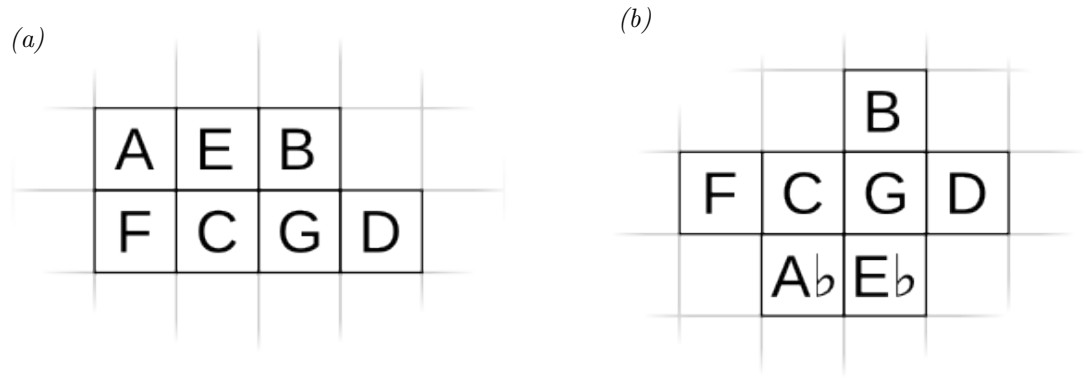

Figure 3: Clusters of points in the tonal space that represent the keys of C major (a) and C harmonic minor (b)

how the chord relates to the current key. This may be *tonic*, *dominant* or *subdominant*, corresponding to the three chords from which we defined the notes of the key (see section 2.1).

In order to understand how the notes of a piece relate to this analysis, it is necessary to form structures analogous to the syntactic structures that in language relate the words of a sentence to their contribution to the sentence's meaning. A dominant chord is inherently incomplete. It calls for its tonic to give a sense of completion, or *resolution*. The pair is characterized by a feeling of tension and resolution. Such pairs of dominant tension and tonic resolution, along with the corresponding subdominant-tonic pairs, are the building blocks of harmonic syntactic structure. The structure is referred to as a *cadence*.

A chord may stand in such a dominant-tonic relationship to another chord which itself is a dominant chord. This chord is known as a *second-level* dominant and the resulting structure as an *extended* cadence. This phenomenon is especially common in jazz harmony, where further levels may be found, forming even more extended cadences.

The structure of an extended cadence can be seen in figure 4a. More complex syntactic structures are formed from these basic building blocks. For example, a structure similar to coordination in language is implicit when a cadence is left unresolved whilst another cadence takes place, before the resolution finally appears. The structure of an instance of this can be seen in figure 4b. Indeed, this structure may itself be hierarchical: before the eventual resolution is reached, another coordination may occur, as in figure 5.

A further characteristic of tonal music is the notion of *substitution*. Given a set of simultaneous notes, not only is their harmonic relation to surrounding chords ambiguous, but one cannot even say what the root of this chord itself is. The root

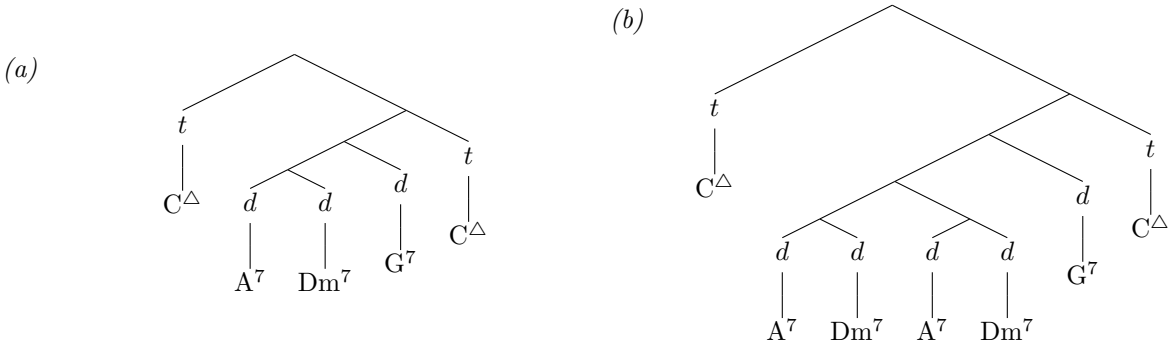

Figure 4: *(a)* The hierarchical structure of an extended cadence. *(b)* An extended cadence with another level of structure: both A^7 Dm^7 passages combine with their resolution G^7.

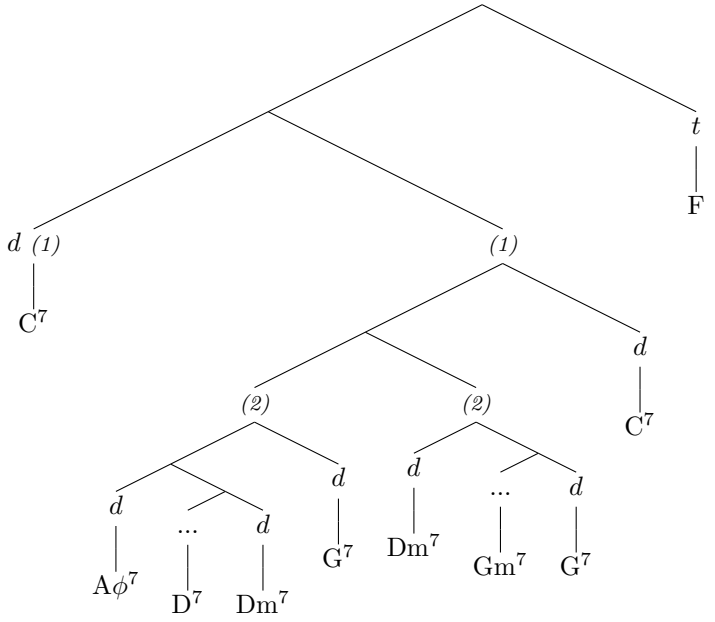

Figure 5: The structure of an extended cadence from *Call Me Irresponsible* with two nested levels of coordination. The coordinated constituents are marked *1* and *2*.

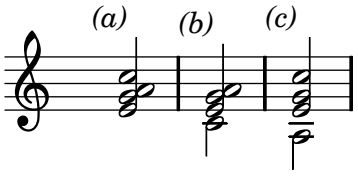

Figure 6: Ambiguity of interpretation of a chord's root. The chord a could be interpreted as C(6) by taking the C as the root (see b), or as Am^7 by taking the A as the root (c).

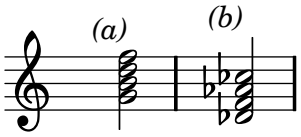

Figure 7: The tritone substitution. A dominant seventh chord in C major (a) followed by its tritone substitution (b), which can serve the same function in the same context.

of a chord is not always voiced as the lowest note (though it frequently is) – an effect known as *inversion*. This means that a given combination of notes can be interpreted in a variety of ways by choosing different notes as the root. Figure 6 shows an example of this. The first chord has two probable interpretations – C(6) or Am^7 – depending on which note is chosen as the root. This gives rise to one form of substitution: a chord could be labeled as C(6) if it is played as in figure 6b, but be correctly interpreted as Am^7, accounting for an inversion.

Another form of substitution is due to the fact that certain chords can be used in place of another chord, on a different root, to serve the same harmonic function. These substitutions are usually also derived from inversion: however, their etymologies are more obscure, as they involve the deletion of many of the notes of the original chords, retaining only certain important notes. Although they are theoretically derived from the chords they substitute for, they tend to be labeled as the chords they resemble for ease of reading.

2.4 Automatic Analysis

The chord roots and functions of a passage of harmony are hugely ambiguous. Nevertheless, a human listener has no difficulty in unconsciously performing the sort of analysis described above while listening to a performance. A large part of the ambiguity of interpretation is a result of performing the music in equal temperament, in which each note maps to a theoretically infinite number of analyses. Further ambiguity is introduced by substitutions, making the interpretation of the chord underlying the heard chord heavily dependent on its context. These ambiguities can be reduced by reference to the harmonic, melodic and rhythmic context of a chord.

If a system can automatically determine the movements through the tonal space underlying a chord sequence or passage of performed music, it is able to address the problems listed in section 1. The problem of score transcription, for example, depends on an analysis of the current key and the relation of a note to that key to determine how the note should be written. The melody in figure 8 would never be written as in 8b by a musician, and the tonal space analysis alongside the transcriptions makes it clear why.

Analysis of key and function are particularly important in jazz music, where performers improvise on the basis of a set form, which often consists of a melody and chord sequence. The analysis is crucial to determine not only what substitutions and chord variants can be used instead of the original chords, but what scales and idioms will work at any particular point.

3 Previous Work

The theory of tonality that we use as the basis for our semantic analysis of harmony, as described in the previous section, was developed by Longuet-Higgins (1979). Steedman (1984) and Steedman (1996) developed the idea of using grammars to perform first the structural analysis, using context-free grammars (CFGs), and later an analysis in the tonal space, using combinatory categorial grammar (CCG, Steedman (2000)). Steedman (1996) described a small grammar of chord sequences designed to generate 12-bar blues sequences. This grammar was heavily limited in its coverage and was unable to handle anything beyond simple cadence structures. Steedman and Wilding substan-

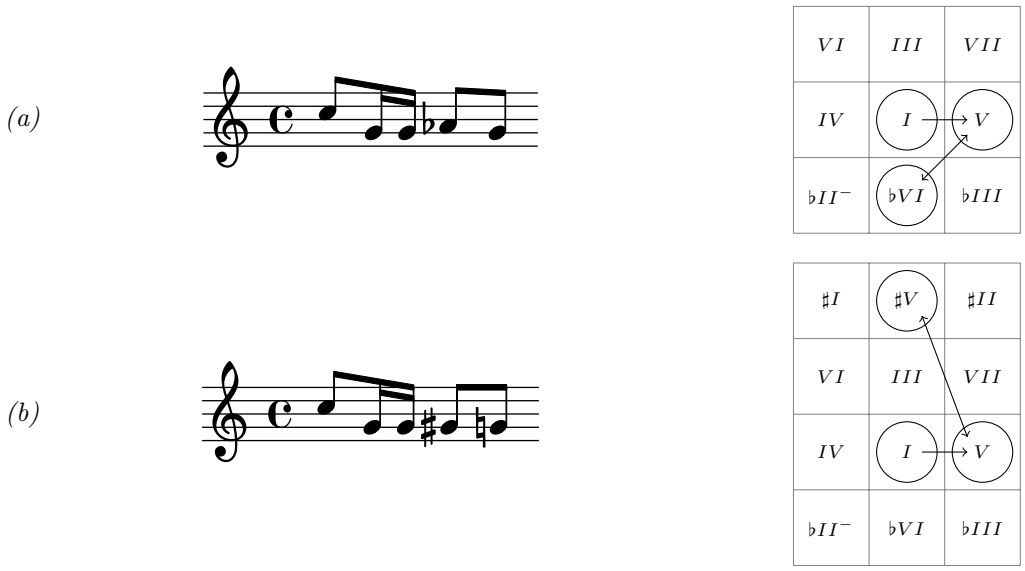

Figure 8: Two different ways of transcribing a simple passage, equivalent under equal temperament. The tonal space reveals why a is preferred over b, which uses a more remote interval under just intonation

tially developed this grammar (Wilding (2007)) so that it was able to interpret a much wider range of chord sequences and more sophisticated harmonic structures.

Many others have used linguistic grammar formalisms to analyse music. Early attempts (such as Roads and Wieneke (1979), Smoliar (1976), Winograd (1968)) experimented with various formalisms, some taken directly from their linguistic setting, others heavily augmented to adapt them to music. Winograd (1968) applies systemic grammar to the problem of harmonic analysis and relates the grammatical analysis to considerations of musical semantics, in particular to recognizing a chord's function in the current tonality.

More recently, Haas et al. (2009) use handcrafted CFGs to parse chord sequences. They apply the resulting parse trees to the task of measuring musical similarity by comparing tree structures.

Another major line of research is that deriving from Lerdahl and Jackendoff (1996). A Generative Theory of Tonal Music (GTTM) followed from Bernstein (1976), a series of talks which attempted to place music theory in the framework of generative grammars. It described a theory of musical analysis made up of a combination of *well-formedness rules*, analogous to a grammar, and *preference rules*, rather like those that later emerged in optimality theory.

A recent development in this line is Katz and Pesetsky (2009), who attempt a re-alignment of GTTM with linguistic theory, claiming that the authors underestimated the extent of a possible alignment between the theories. They reformulate GTTM in terms of X-bar theory. Others have continued to build on GTTM in recent years. For example, Hamanaka et al. (2006) implemented the theory, with some modifications.

Temperley and Sleator (1999) have a theory of musical analysis based on preference rules similar to those proposed in GTTM. They have implemented a system for metrical and harmonic analysis using the hand-crafted rules.

4 Our Approach

Our current approach is the direct continuation of Steedman (1996) and Wilding (2007). We aim to build a relatively wide-coverage CCG grammar for chord progressions, and eventually for streams of performance data, with a parsing model permitting practical application of the theory to real music processing problems. We focus for now on a particular domain, that of jazz standards. We hope to be able to generalize this to other dialects of Western tonal music.

The approach is founded on the music-language analogy by learning from the theoretical developments and discoveries in computational linguistics and applying techniques used in language processing to what we see as analogous problems in music processing. Although our work is formally distant from GTTM and its descendents, in this much the approach we take is in the spirit of Bernstein (1976). The techniques drawn from natural language processing include statistical parsing and probabilistic modeling techniques.

Unlike GTTM and related work, we analyse the syntactic structure of music in order to produce an interpretation of it in an abstract domain. This is Longuet-Higgins' tonal space and we believe that the interpretation in this domain can reveal important semantic characteristics of the music, not unlike those musical features that Bernstein defines as "the meanings of music". Indeed, it is necessary to perform this sort of interpretation of music before Bernstein's transformational musical semantics become explicit.

Our analytical process, implemented in the parser, produces an interpretation at the level of chords and notes in Longuet-Higgins' tonal space. This interpretation can be used for the applications suggested in section 1 as explained in section 2.

Just as for language, a syntactic analysis of a surface form represents the structure that maps the surface form to its meaning. Such structure exists in various largely orthogonal aspects of music, but we focus particularly on harmonic structure. Another feature of music that exhibits a hierarchical structure is meter. Although there is a strong influence of metrical structure on harmonic interpretation, a human listener is able in many cases to interpret harmony even in the absence of any metrical cues. We therefore separate the two processes for now and consider harmonic analysis on its own.

The ambiguity of interpretation of music is similar to ambiguity found in natural language. We use the paradigm of the handling of ambiguity in language as the basis for dealing with that in music. There are important differences in the balance of types of ambiguity, however. For example, most natural languages have a wider variety of syntactic structures to consider than those found in music, but features of a word on its own narrow down its possible interpretations far more than a group of notes.

5 The Grammar

5.1 Theory

Our grammar uses a modification of CCG as it is applied to natural language to represent the syntactic structures of music. For an introduction to CCG for language, see Steedman (2000).

5.1.1 Basic Syntax

The syntactic categories of our grammar express some limited information about the tonality of the passage, though only enough information to know how adjacent passages may be interpreted alongside it. The category notation uses symbols that look like the roman numeral notation used to transcribe chords relative to a key. It is important to note, however, that the roman numerals do not in this case imply a harmonic analysis. A category containing the symbol I should not be confused for the tonic note or chord of the current key. In fact, it refers to the (equally tempered) pitch class I relative to the chords of the input.

This notation allows us to define root interpretations and resolutions irrespective of the absolute pitch of the input. For example, the category for dominant chords described below includes the symbol I, which represents the root of the chord as it is played (not the I of the current key), and the symbol IV, representing a IV relative to the chord.

Tonic chords and passages of tonic chords receive atomic categories, reflecting their fully resolved nature. Atomic categories in our grammar have the form $A–B$, where A and B are roman numeral symbols. The roots convey a notation of the true root of the chord, though not as yet freed from the equal temperament of the input. The category $A–B$ denotes that the span it covers begins at the root A and ends at B.

The simplest lexical category in the grammar interprets unsubstituted tonic chords: $I–I$. This rather trivially states that the one-chord passage begins and ends on the root I.

Dominant chords receive a complex category, indicating that they need to combine with a particular resolution immediately following them. The simple, unsubstituted dominant category in the lexicon is $I–X \,/\, IV–X$. The root I is the played root of the chord and the IV is its resolution, a perfect fifth below. To be combined in a derivation, the category needs to be followed by a passage interpreted as beginning on the root a perfect fifth below it and the result will begin, of course, on the root of this chord. The eventual root of the whole passage, that represented by X, is unimportant.

More lexical categories are described below in section 5.2.

5.1.2 Interpretation of Cadences

Using this syntax, we are able to interpret the resolution of a cadence simply by applying the cadence category (which has a forward slash) to the tonic category of its resolution.

$$\dfrac{\dfrac{\texttt{V7}}{V–X \,/\, I–X} \quad \dfrac{\texttt{I}}{I–I}}{V–I}\,{>}$$

Extended cadences, as described in section 2.3 may be derived by successive application of dominant categories.

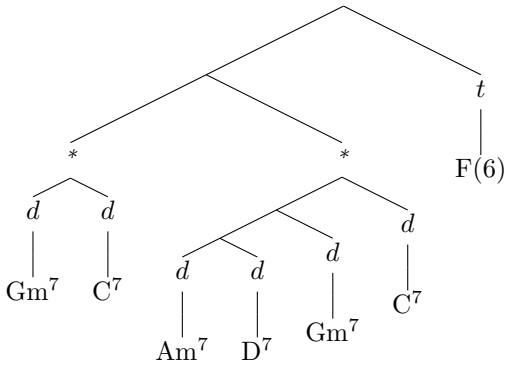

Figure 9: An extended cadence with coordination, from *Bluesette* (transposed). The coordination of a short partial cadence with a longer one rules out requiring coordinated constituents to share their start point. Instead we only require them to share a resolution.

$$\frac{\texttt{IIm7}}{IIm\text{–}X \;/\; I\text{–}X} \qquad \frac{\dfrac{\texttt{V7}}{V\text{–}X \;/\; I\text{–}X} \qquad \dfrac{\texttt{I}}{I\text{–}I}}{V\text{–}I}\,{>}}{IIm\text{–}I}\,{>}$$

5.1.3 Initial Function Feature

An additional piece of information required to express the syntactic limitations on interpretation is the function of the chord at the start of the span. We introduce this extra feature into the notation for syntactic categories. A span that begins with a dominant chord will have the value 'D' for this feature, whereas one that begins on a tonic will have the value 'T'. Similarly, one beginning with a plagal cadence will have 'S' (for subdominant). We write the feature above each atomic category: $A^{\alpha}_{-}B$. For example, $III^{D}_{-}I$ spans the tonal centres III and I and begins with a dominant step.

The purpose of this is to enforce a sort of agreement of cadences (a bit like agreement in language). Several dominant chords resolving to one another may form a single authentic cadence and subdominant chords may similarly form a plagal cadence.

The simple dominant category described above therefore now looks like this:

$$I^{D}_{-}X \;/\; IV^{DT}_{-}X$$

That is, the argument may begin with a tonic, if this is a full resolution, or a dominant, if we are continuing in an extended cadence, but not a subdominant. The result of application will begin with a dominant chord, this chord.

5.1.4 Stringing Cadences Together

We are now able to build up extended cadences and apply them to their resolution. However, most pieces of music are made up of more than one cadence.

To produce a derivation of multiple consecutive cadences, we introduce a new grammatical rule called *development*. It allows any atomic (i.e. resolved) category to combine with another resolved passage following it.

$$X^{\alpha}_{-}Y \qquad Z\text{–}W \qquad \Rightarrow \qquad X^{\alpha}_{-}W$$

The categories are not required to be in the same key: we may very well modulate to a new key via a cadence and do not wish to restrict ourselves always to return to the same tonic.

5.1.5 Coordination

We use one further grammatical rule, which we call *coordination*, because of its structural similarity to coordination in natural language. The phenomenon of coordinated cadences was mentioned in section 2.3. A cadence may not proceed directly by dominant (or subdominant) chords to its resolution, but may leave a tension temporarily unresolved. There may be a repetition of some steps of the cadence, perhaps in a substituted form, or even a longer extended cadence before the resolution of that hanging chord is reached. (See the trees of figures 4b and 5 for examples.)

A coordination grammatical rule combined unresolved fragments of cadences into a single constituent so that they share the eventual resolution. A series of unresolved cadence steps is first combined into a constituent using composition. The rule takes the form

$$X/Z \;\; Y/Z \Rightarrow_{\&} (X/Z).$$

The two categories are not required to be identical, but only to seek the same resolution (their

T. $\quad \text{X(m)} := I(m) \overset{T}{-} I(m) \quad : \; I_T :: Nil$

D. $\quad \text{X(m)}^7 := I(m) \overset{D}{-} Y(m)_1 \; /_A \; IV(m)_2 \overset{DT}{-} Y(m)_1 \quad : \; \lambda x. I_D :: x$

D_Tt. $\quad \text{X(m)}^7 := \flat V(m) \overset{D}{-} Y(m)_1 \; /_A \; VII(m)_2 \overset{DT}{-} Y(m)_1 \quad : \; \lambda x. \flat V_D :: x$

S. $\quad \text{X(m)} := I(m) \overset{S}{-} Y(m)_1 \; /_P \; V(m)_2 \overset{ST}{-} Y(m)_1 \quad : \; \lambda x. I_S :: x$

Rep. $\quad \text{X(m)} := I(m) \overset{T}{-} Y(m)_1 \; / \; I(m)_2 \overset{T}{-} Y(m)_1 \quad : \; \lambda x. x$

Pass_VI. $\quad \text{X}\circ 7 := VI(m) \overset{T}{-} Y(m)_1 \; / \; VI(m) \overset{T}{-} Y(m)_1 \quad : \; \lambda x. x$

TC_IVR. $\quad \text{X(m)} := (\; A(m)_1 \overset{\alpha}{-} B(m)_2 \; / \; V(m)_3 \overset{T}{-} B(m)_2 \;) \backslash \; A(m)_1 \overset{\alpha}{-} V(m)_4 \quad : \; \lambda x, y. x + y$

Figure 10: A selection of categories from the Jazz Grammar lexicon. When applied to a chord, the roots in a category are transformed to be relative to the root of the chord (denoted X here). (m) represents a minorness dependent on the chord's minorness (no subscript) or bound to be equal to similarly-subscripted minors by unification.

argument). This may seem an odd decision in the light of an example like that in figure 5. In this case, it might seem more reasonable to combine the (D^7 Dm^7 G^7) phrase with (Dm^7 Gm^7 G^7), which (but for minorness) can have identical categories. This interpretation could be produced by a less permissive rule $X/Z \quad X/Z \Rightarrow_{\&} (X/Z)$. Figure 9 demonstrates that this is too strict a condition for coordination. In this case, the latter phrase is longer than the former, which consists of only two chords. The two could not be coordinated by the stricter rule, since the first partial cadence does not include the initial chords of the second, so cannot begin at point A.

In this form, the rule is not sufficiently constrained. We only wish it to be able to combine constituents that consist purely of cadence steps. Furthermore, these two unresolved cadences must be of the same type: we do not wish to coordinate a plagal cadence with an authentic cadence. These constraints are imposed by a system of slash modalities. Each slash is annotated with an 'A' if it represents a pure unresolved authentic cadence and a 'P' for a plagal cadence. Some simple extensions to the other grammatical rules ensure that the markers are correctly propagated through the derivation.

Our coordination rule is now

$$X/_m Z \quad Y/_m Z \quad \Rightarrow_{\&} \quad (X/_m Z), \quad \textit{iff } m \in \{A, P\}$$

5.1.6 Semantics

Each lexical item is endowed with a functional logical form, expressed using the lambda calculus. Literals in the logical forms represent points in the Longuet-Higgins tonal space. During a derivation the logical forms are combined. The resulting logical form on a sign spanning a full chord sequence is a list of points forming a path through the tonal space.

Lists are expressed using the standard functional list structure, in which a list may be either a head element joined to another list (its tail) using the *cons* operator (written ::), or the empty list (written Nil). The tonal space points that are the elements of the lists contain two pieces of information: the chord root as a the name of a point in the space; and the function of the chord (T, D or S).

Thus, the semantics of the obvious interpretation of the standard turnaround `IM7 IIm7 V7 IM7` would be $I_T :: II_D :: V_D :: I_T$.

5.2 Lexicon

We present here a representative sample of the current grammar's lexicon. The lexicon is large and repetitive, so is not reproduced in full here. The fragment of the lexicon is shown in figure 10.

The category named 'T' is the simple tonal interpretation introduced in section 5.1.1. It now has a semantics: a path with a single point in the tonal space. 'D' is the simple dominant chord interpretation, also described in 5.1.1. Its semantics prepends a point in the tonal space, marked as dominant, to the path that follows it (beginning with its resolution).

'D_Tt' interprets a chord as a dominant chord that has undergone a tritone substitution. Its true root is considered to be $\flat V$ (or $\sharp IV$ – these are equated by equal temperament) relative to its played root. Consequently, its resolution is expected to be a fifth below its true root – that is, a semitone below its played root. The grammar

contains many categories to handle an array of substitutions, of which this is the most common. Category 'S' is the plagal equivalent of 'D'.

'Rep' effectively ignores a chord where it is followed by another chord on the same route by allowing them to combine and take the semantics of the second chord.

'Pass_VI' applies to diminished seventh chords, which under one interpretation are nothing more than *passing chords* – a chord of simultaneous passing notes which serve only to interpolate between the surrounding chords. It consequently contributes nothing to the semantics.

'TC_IVR' is the most elaborate category in the grammar and, despite its appearance, does not do anything very interesting. It is a tonic colouration chord. It is applied to the `IV` in a common elaboration of a tonic passage on I: `I IV I`. It simply concatenates the paths that precede and succeed it and contributes nothing to the path itself.

6 Baselines

6.1 Models

We have implemented two simple statistical parsing methods. These serve as an exploration of the modeling problem and provide baseline results for us to improve on with more sophisticated models.

6.1.1 C&C Supertagger

Clark (2002) describes a statistical parsing method for CCG using a supertagger. Parsing becomes infeasible when many possible categories are assigned by the grammar to each token in the input. A supertagger models the probability of assigning categories to the input on the basis of the immediate context. It uses the model to restrict the categories assigned to each token, aiming to eliminate very unlikely interpretations at the lexical level using features of the short-range context.

Curran et al. (2007) implemented a fast parser (the C&C Parser) using a supertagger component with a log-linear model trained on tagged training data. We have used the C&C supertagger component directly. We trained the supertagger on observations from annotated chord sequences in our corpus. In order that the model should be insensitive to the absolute pitch of chords, the observations consisted of a numeric interval in semitones between two chords and the chord type of the first chord. Once the observations in this format have been extracted from the corpus, along with the corresponding annotated tags, the supertagger can be trained directly on this data.

We parse sequences with our Jazz Parser, which calls the C&C supertagger to get a distribution over tags for each chord. It does this by calling the C&C supertagger with a beam threshold of 0, so that it returns all the tags to which it assigns any probability. The parser first adds just the most probable tags to the chart and attempts to parse. If no full parse is possible it adds further lexical signs of lower probability until a full parse results. The parser has its own parameter to decide how to split up the tags assigned to each chord to add them to the chart. This is a beam defining the maximum ratio between the highest and lowest probability added in one batch. We arbitrarily set this value to 0.8. Note, though, that adjusting this would not affect the recall of the parser, since it will use all the tags returned by the tagger if it finds no parse.

The parser ranks the results of the parse by the combined probability of all the tags that contributed to the derivation.

6.1.2 PCFG

Hockenmaier (2001) describes the application of a simple statistical parsing model based on probabilistic context-free grammars (PCFGs). Like PCFGs, the model estimates probabilities at each node of the derivation tree of the generated daughters, conditioned on the parent. In this way, for each tree the probability that that tree generates the observed data is estimated. The probabilities are estimated from the frequency of expansions in a corpus fully annotated with tree structures.

A feature of CCG is that the same parse can be produced from the same lexical categories by different derivations. This presents a problem when training the model on a single derivation tree for each sentence in the training set. We employ the solution used by Hockenmaier (2001), which is to define a scheme for producing a single canonical tree for any derivation and to train only on this tree.

Our corpus encodes with minimal annotations a canonical tree for each chord sequence[1]. We train a simple frequentist model on our jazz treebank. For this baseline model, unseen words (if any) and expansions are handled using add-one smoothing.

[1] Precisely, the only information required beyond the categories themselves to determine the full tree structure is markers of where coordination occurs.

	C&C supertagger	PCFG
Recall	21.4%	100.0%
Mean accuracy	90.8% (10.0%)	59.8% (11.9%)
Mean error rate	0.10 (0.11)	0.41 (0.13)

Table 1: Evaluation of analysis from top results against gold standard analysis, for the two baseline models. The supertagger model parses few sequences successfully, but performs very well when it does. The PCFG model produces an analysis for every sequence, but makes about 2 errors in every 5 chords. Figures in brackets are standard deviations.

6.2 Results and Analysis

Currently, our small corpus contains only a labelled training set. To get an initial idea of how these models will perform, we evaluated them using ten-fold cross-validation on this training set.

It is not clear what is a good evaluation metric for our data. The eventual output of the parser that we are interested in is the tonal space analysis of the music. We should therefore ideally evaluate the similarity between the tonal space analysis produced by the parser and that annotated in the gold standard.

In order to do this, we define a distance metric between two tonal space paths, which are represented as lists of points. The metric is similar to the word error rate metric used to evaluate machine translation systems. We calculate the Levenshtein distance (edit distance) between the sequences of points, assigning a cost of 1 to each deletion, insertion and error, and divide this by the number of points in the gold standard path. We refer to this measure as the *point error rate* (PER). It can be thought of as the average number of errors per point in the gold standard.

We make a further elaboration to this metric. Recall that the analysis contains two pieces of information about each point: the chord root and the harmonic function. Each of these is an independently valuable piece of information and we would not like to penalize too harshly a model that, for example, correctly analysed the functional structure, but misinterpreted the chord roots. We therefore assign a cost of 0.5 to errors in which only one of these features is incorrect. It is also interesting to report the proportion of errors for which a model correctly analysed only one of these.

$$PER = \frac{D + I + E_b + \frac{E_r + E_f}{2}}{|P_g|}$$

D: # deletions
I: # insertions
E_r: # root errors
E_f: # function errors
E_b: # full errors
P_g: gold standard path

We report a second measure which is a form of accuracy and which we refer to as *path accuracy* (PA). We compute the distance of the optimal alignment between the top result path and the gold standard path as above, but rather than dividing it by the length of the gold standard, divide by the greater of the lengths of the paths. We then substract this from 1. The result ranges from 0, in the case where no alignment was made between the two paths, to 1, if they were perfectly aligned.

$$PA = 1 - \frac{D + I + E_b + \frac{E_r + E_f}{2}}{max(|P_g|, |P_r|)}$$

P_r: path of the top result

Our corpus currently contains about 100 chord sequences. Only about 60 of these are fully annotated – the others contain holes of, in most cases, one or two chords, whose interpretation we are uncertain of or unable to handle with the grammar in its current state. For the time being, we simply ignore sequences that are not fully annotated.

In table 1 we report results based on our distance metric for the top analysis produced by each model in a ten-fold cross validation.

The supertagger model had a very low recall. The chord sequences in the corpus vary in length from 20 chords to 95. The supertagger model does well at assigning the common categories accurately. It performs very well on structurally simple sequences containing only these categories. On sequences with rare categories, the supertagger never produces a sequence of tags that leads to a full parse. Figure 11 shows the distributions of tags in the corpus and of those that the supertagger assigns with highest probability. The supertagger only ever picks one of the most frequent 11 tags.

The PCFG model succeeds in finding a full parse for every sequence. Its accuracy is low, again because of the coverage of the training data. Figure 12 shows the proportion of substitutions in the alignment with the gold standard that got either the root or function correct. This reveals that in only very few cases did the model fail to identify either.

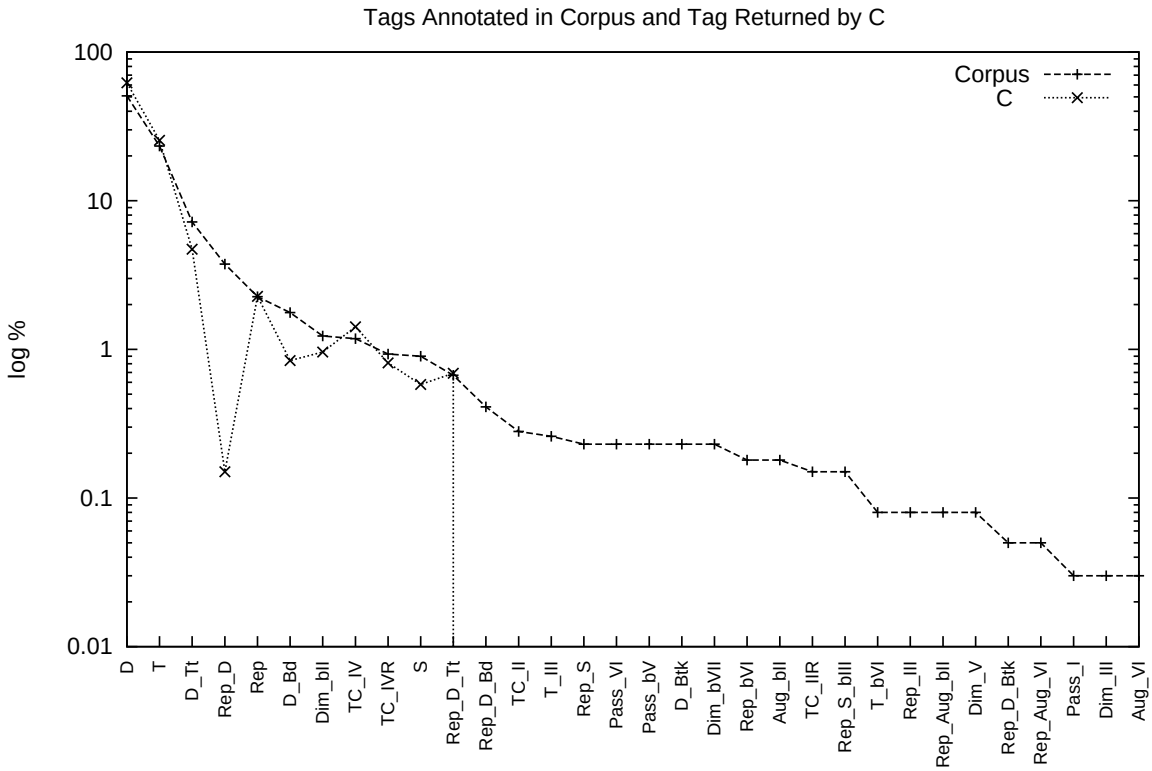

Figure 11: The distribution over lexical categories seen in the annotations and in the most probable tags assigned by the supertagger. After the top 10, the training data drops to almost no examples. The supertagger's model fits the distribution closely and returns no tags after the most frequent 11.

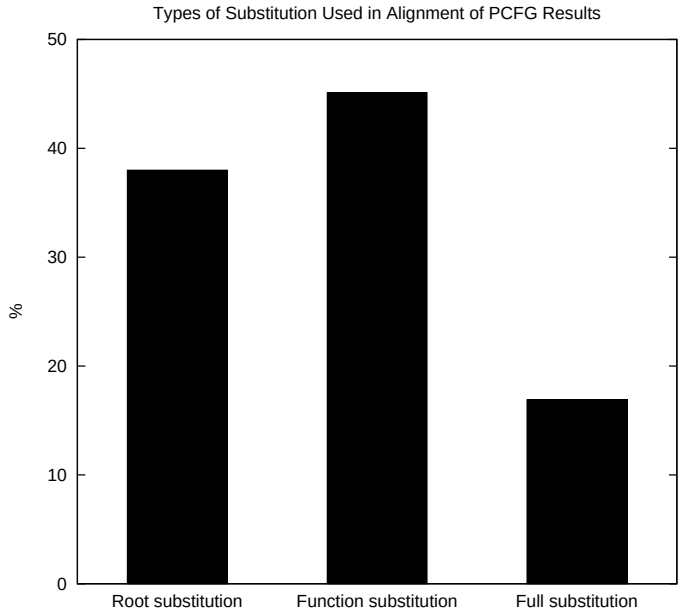

Figure 12: The distribution of types of substitution observed in alignments of the PCFG's top results to gold standard tonal space paths. The model rarely misinterprets both the root and the function.

The recall of the supertagger-based parser and the accuracy of the PCFG are unsurprising given the sparsity of the training data, in which many tags only appear a few times. Most of the corpus is made up of the common, easily predicted categories and the supertagger, as expected, labels these well. Over the whole corpus it assigns the highest probability to the tag that was labeled in the gold standard in 80.6% of cases. However, this initial experiment demonstrates that this is not sufficient: if the correct category is not allowed by the tagger for one chord, the whole sequence is unparsable. The PCFG model also suffers from the data sparsity. The current model uses only very simple smoothing of unseen expansions. We expect to be able to get a large improvement in the results by using better smoothing techniques.

7 Work So Far

The work carried out so far on this project built directly on the work of Wilding (2007). Wilding (2007) developed the small blues chord sequence grammar of Steedman (1996) into a larger grammar designed to generate chord sequences of jazz standards. He also built a parser that could use this grammar to process chord sequences, produce analyses in terms of cadences and test these analyses against predefined specifications intended to capture the characteristics of common harmonic forms.

During this year, we have made significant further developments of the formalism for expressing musical syntactic categories (a modified version of the standard CCG category notation for natural language). This has resulted in the grammar formalism and lexicon outlined in section 5. The modifications make it possible to describe more of the sorts of syntactic structures found in music. We have also substantially improved the way the semantics of tonal space movements is described and composed during derivations.

Additions to the lexicon of the jazz standards grammar have increased the coverage of the grammar so that it is now able to provide interpretations of a far wider range of chord sequences in the domain. The grammar is now able to produce a tonal space interpretation of many sequences taken from the Real Book (Hal Leonard Corp. (2006)).

We have built a corpus of chord sequences by manually annotating sequences with grammatical information under the current jazz standards grammar. The corpus implicitly contains a full canonical derivation tree and semantic interpretation for each chord sequence. The corpus contains about 100 chord sequences and about 4,000 chords. It is intended for use as training data for statistical parsing models.

We have implemented two statistical parsing models, following the current two leading statistical parsing techniques with CCG for natural language. The models are very simple and are intended as a proof of concept and a baseline for the development of more sophisticated models. The first is a log-linear model for supertagging, after Clark (2002). It is implemented by using the C&C supertagger (Curran et al. (2007)) directly to assign grammatical categories to chords at the front end of the parser. The second is a generative probabilistic parsing model, similar to a PCFG, after Hockenmaier (2001). This is built into the parser and estimates its probabilities from the tree structures in the corpus. Both of these models have been built into the existing parser developed by Wilding (2007). We described the models in section 6.1. We have extended the parser to make it easy to build future alternative models into it. Some preliminary baseline results are given in section 6.2.

8 Proposed Work

Having now established a baseline for the parsing of chord sequences using straightforward statistical parsing techniques, we intend to improve on these results with more sophisticated statistical models. The unimpressive results reported for the baseline models are unsurprising, since they are naively constructed and trained on only a small amount of data.

There are several possibilities to explore for immediate improvement. Currently the PCFG model using only very naive smoothing and the use of suitable smoothing techniques is critical, given the sparsity of our dataset. The current PCFG model also uses a simple lexical generation probability estimate and this could be improved by making use of features of the immediate context of the sort that influence the supertagger model. The PCFG model may also be improved by using a more informative notion of the head of a syntactic category. We expect to be able to make some improvement on the reported results for the C&C supertagger easily, by simple adjustments to its parameters and the way the parser uses the supertagger's output. We will then investigate using similar lexical models in ways better suited to musical processing and combining this model with the PCFG parsing model.

Our current dataset constitutes only a training set. So far we have used cross validation on partitions of this set to estimate an upper bound on parsing accuracy. However, to produce meaningful

results, we clearly need a separate annotated test set. We propose to construct such a set by annotating a further set of chord sequences in the same way as the training set. We will need to take care to select an unbiased sample of chord sequences to include in this set. We will also need to consider what is a suitable size for the set, since annotation is a very time-consuming process.

We hope that we may be able to handle rare grammatical interpretations by employing better smoothing techniques, but it may turn out that the data sparsity is too great. In this case, we will need to consider semi-supervised models, an area we have not yet investigated.

An alternative solution is to consider other annotation schemes. The current annotation of derivation trees (and implicitly tonal space interpretations) is time consuming and can only be done with an intimate understanding of the chord grammar. A more abstract annotation, such as simply chord roots, or chord roots and harmonic functions, may be sufficient for the model to learn from and would allow faster preparation of training data by more annotators.

It is still not clear what evaluation metrics are most suitable for this task. Currently, we report results for similarity between the tonal space path of the output and the gold standard tonal space path, using a metric similar to word error rate (that is, edit distance average over the path's points). It is important for future exploration of alternative solutions to the problem that the evaluation metric is not bound to this specific solution. Our eventual aim is to extend our models to handle not only strings of chord sequences, but actual musical input (such as MIDI events). Our current metric would not generalize well to other forms of input such as this.

An important measure of the difficulty of the task and the expected ceiling result is human annotator agreement. So far we have not measured agreement between different annotators, since there has been only one annotator. A more abstract form of annotation could have other benefits, mentioned above, but would also allow us to get annotations from more people and to measure agreement. We expect agreement to be low, due to the great ambiguity of interpretation in music.

A large array of problems will face us when we come to consider how to handle MIDI-type input directly. This generalization is necessary for many of the applications we list in section 1. Ideally, we would like a generative model to incorporate the full realization of music, rather than just a grammatical model of chord labels produced by a separate process of segmentation and labelling. The "bag of notes" assumption made by many models is a poor approximation for a parsing model and fails to provide a full generative model of music production. A realization model needs to take account of the interaction of harmonic prominence of notes with metrical structure, as well as idiomatic forms, such as scales and arpeggios.

If we succeed in building a model of realization that performs sufficiently well, we intend to implement some of the applications suggested in section 1. Of particular interest are the evaluation of musical similarity, realization of a piece in just temperament and generation of fully realized accompaniments.

9 Plan

The following plan includes the broad tasks outlined in section 8, with the months in which we hope to tackle each task.

1. Devise evaluation methods
    - use them to do consistent evaluation of baseline methods already implemented
    - evaluate improvement of better models as they are developed

    10-11
2. Enhance the baseline models with more sophisticated smoothing 11-13
3. Consider alternative annotation schemes. Possibly get more data 10-14
4. Try more sophisticated statistical models 14-20
5. Investigate annotator agreement. Experiment with other annotators 14-20
6. Investigate semi-supervised methods 20-24
7. MIDI realization models 20-34
8. Use best full models (tonal space interpretation from MIDI) to implement some applications 25-34
9. Write up 31-36

References

M. Allan. Harmonising chorales in the style of Johann Sebastian Bach. Master's thesis, School of Informatics, University of Edinburgh, 2002.

M. Baroni, S. Maguire, and W. Drabkin. The concept of musical grammar. *Music Analysis*, 2(2): 175208, 1983.

L. Bernstein. *The Unanswered Question: Six Talks at Harvard*. Harvard University Press, 1976.

S. Clark. Supertagging for combinatory categorial grammar. *Proceeding of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks*, pages 101–106, 2002.

D. Conklin and I.H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

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

J. Curran, S. Clark, and J. Bos. Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

W. B. Haas, M. Rohrmeier, R. C. Veltkamp, and F. Wiering. Modeling harmonic similarity using a generative grammar of tonal harmony. In *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, pages 1–6. International Society for Music Information Retrieval (ISMIR), 2009.

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

M. Hamanaka, K. Hirata, and S. Tojo. Implementing A Generative Theory of Tonal Music. *Journal of New Music Research*, 35(4):249–277, 2006.

H. Helmholtz. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Dover Publications, 1885.

J. Hockenmaier. Statistical parsing for CCG with simple generative models. In *Association for Computational Linguistics 39th annual meeting and 10th conference of the European Chapter: July 9th-11th 2001: CNRD-Institut de Recherche en Informatique de Toulouse, and Université des Sciences Sociales, Toulouse, France*, volume 39, pages 7–12, 2001.

J. Katz and D. Pesetsky. The identity thesis for language and music. In draft: lingBuzz/000959, 2009.

S. Koelsch, T. C. Gunter, M. Wittfoth, and D. Sammler. Interaction between syntax processing in language and in music: An ERP study. *Journal of Cognitive Neuroscience*, 15: 1565–1577, 2005.

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

H. C. Longuet-Higgins. The perception of music. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 205(1160):307–322, 1979.

A. D. Patel. Language, music, syntax and the brain. *Nature Neuroscience*, 6(7):688–691, 2003.

D. Ponsford, G. Wiggins, and C. Mellish. Statistical learning of harmonic movement. *Journal of New Music Research*, 28(2):150–177, 1999.

J.P. Rameau and P. (tr.) Gossett. *Treatise on Harmony*. Dover Publications, 1971.

C. Roads and P. Wieneke. Grammars as representations for music. *Computer Music Journal*, 3 (1):4855, 1979.

S. Smoliar. Music programs: An approach to music theory through computational linguistics. *Journal of Music Theory*, 20(1):105–131, 1976.

M. Steedman. The blues and the abstract truth: Music and mental models. In A. Garnham and J. Oakhill, editors, *Mental Models in Cognitive Science*, pages 305–318. Erlbaum, 1996.

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

M. Steedman. A generative grammar for jazz chords sequences. *Music Perception*, 2:52–77, 1984.

D. Temperley and D. Sleator. Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27, 1999.

M. Wilding. Automatic harmonic analysis of jazz chord progressions using a musical categorial grammar. Master's thesis, School of Informatics, University of Edinburgh, 2007.

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