Advances in Multiple Viewpoint Systems and Applications in Modelling Higher Order Musical Structure

by

Thomas Hedges

A thesis submitted to the University of London for the degree of
Doctor of Philosophy

Department of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

July 2017

I, Thomas Hedges, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature:

Date:

20th July, 2017

Details of collaboration and publications:

Chapter 5 has been published in its entirety (see Hedges & Wiggins, 2016b). Chapter 6 contains material previously published in Hedges and Wiggins (2016a). Very preliminary work relating to Chapters 10 and 11 has been presented as Hedges and Wiggins (2015). Small differences in results between previously published work and the current thesis may be accounted for by small corrections in the corpora, subtle changes in the parametrisations, and very minor bug fixes in the implementations. For further publications by the author, see Author's Publications (p. 316).

To Emma

Abstract

Statistical approaches are capable of underpinning strong models of musical structure, perception, and cognition. *Multiple viewpoint systems* are probabilistic models of sequential prediction that aim to capture the multidimensional aspects of a symbolic domain with predictions from multiple finite-context models combined in an information theoretically informed way. Information theory provides an important grounding for such models. In computational terms, information content is an empirical measure of compressibility for model evaluation, and entropy a powerful weighting system for combining predictions from multiple models. In perceptual terms, clear parallels can be drawn between information content and surprise, and entropy and certainty. In cognitive terms information theory underpins explanatory models of both musical representation and expectation.

The thesis makes two broad contributions to the field of statistical modelling of music cognition: firstly, advancing the general understanding of multiple viewpoint systems, and, secondly, developing bottom-up, statistical learning methods capable of capturing higher order structure.

In the first category, novel methods for predicting multiple basic attributes are empirically tested, significantly outperforming established methods, and refuting the assumption found in the literature that basic attributes are statistically independent from one another. Additionally, novel techniques for improving the prediction of *derived viewpoints* (viewpoints that abstract information away from whatever musical surface is under consideration) are introduced and analysed, and their relation with cognitive representations explored. Finally, the performance and suitability of an established algorithm that automatically constructs locally optimal multiple viewpoint systems is tested.

In the second category, the current research brings together a number of existing statistical methods for segmentation and modelling musical surfaces with the aim of representing higher-order structure. A comprehensive review and empirical evaluation of these information theoretic segmentation methods is presented. Methods for labelling higher order segments, akin to layers of abstraction in a representation, are empirically evaluated and the cognitive implications explored. The architecture and performance of the models are assessed from cognitive and musicological perspectives.

Acknowledgments

I owe my sincere thanks to a number of individuals, groups, and institutions that have supported me in varied, but essential, ways throughout the years I have been working on this thesis. Firstly, I would like to thank my primary supervisor, Geraint Wiggins, for his expert guidance, insightful knowledge, and heartfelt support. Also, to my supervisory panel of Simon Dixon and Elaine Chew who have provided excellent feedback at crucial times throughout the PhD programme, and to my examiners Alan Marsden and Bob Sturm for their careful reading of this thesis and thoughtful comments. I have been fortunate to work in fruitful and inspiring environments prompting countless stimulating conversations. Therefore, I owe my thanks to Jamie Forth, Kat Agres, Sascha Griffiths, Stephen McGregor, Mariano Mora-McGinity, Nicholas Harley, Max Droog-Hayes, Callum Goddard, Olson Wolf, Vincent Akkermans, and Sue White as members of the Computational Creativity Lab at Queen Mary University of London, and to François Pachet and Pierre Roy of Sony CSL, Paris. I would like to extend my gratitude to Marcus Pearce for detailed feedback and discussions concerning multiple viewpoint systems, and for his IDyOM implementation that this thesis builds upon. Lastly, I would like to thank Martin Rohrmeier as an excellent mentor through my early stages in academia and research.

On a personal level, I am grateful to my close friends for keeping my spirits high in trying times, to University of London Canoe Polo Club for laughter, paddling, and medals, and to my family for their unwavering support. Finally, I am eternally grateful to my partner, Emma, for her loving support through good times and bad.

Financially, this work was made possible by the support of the Media and Arts Technology programme, EPSRC Doctoral Training Centre EP/G03723X/1.

Contents

List of Figures

List of Tables

List of Abbreviations

ARHMM auto-regressive hidden Markov model

CHARM Common Hierarchical Abstract Representation for Music

CKY Cocke–Younger–Kasami

CTW Context Tree Weighting

DBN Dynamic Bayesian Network

DIC Directed Interval Class

EEG electroencephalogram

EM Expectation-Maximisation

EMG electromyography

ERAN early right anterior negativity

ERP event-related potential

GCT General Chord Type

GIS Generalised Interval System

GTTM Generative Theory of Tonal Music

HMM Hidden Markov Model

IDyOM Information Dynamics of Music

IDyOT Information Dynamics of Thinking

IR Implication-Realisation

IOI inter-onset interval

LTM long-term model

MIDI Musical Instrument Digital Interface

MIR music information retrieval

pcset pitch class set

PPM Prediction by Partial Match

PST Prediction Suffix Tree

OOI outer-onset interval

RBM Restricted Boltzmann Machine

STM short-term model

2TBN two-slice temporal Bayesian net

Part I

Overture

Chapter 1

Introduction

1.1 Motivations

How much can one learn by simply counting occurrences of events in a stream of information? Counting only individual occurrences may give a crude indication of structure, whilst keeping track of the frequency of events over time as 'cause-and-effect' patterns might improve one's structural understanding of the sequence. Further improvements may be found by extending backwards the 'cause' of the cause-and-effect to give the 'effect' a better informed context, or by identifying correlations with other streams of information. However, fairly substantial modifications to the counting strategy would be required to understand perfectly ordinary human phenomena such as a natural languages or musical structure, both of which require structural dependency between non-adjacent events. For example, consider the sentence: 'the *cat* sat on the mat *is* hungry,' and 'the *cats* sat on the mat *are* hungry.' Intuitively, it is immediately obvious why simply counting adjacent words would be an insufficient strategy to learning such a language; the embedded verb phrase 'sat on the mat' may potentially be substituted for an infinite number of other verb phrases.

The informal counting strategies and modelling techniques described above are essentially statistical models of increasing sophistication. A *statistical learning* approach to cognition argues that structure in sequential information can be found purely by counting occurrences and co-occurrences of symbols in the data itself, no additional external domain-specific knowledge (or at least, a minimal amount) is required. The current state of the art in statistical models of cognitive processes occupy a space above simply identifying cause-and-effect patterns (or local dependencies), but below being able to fully account for non-local (or higher order) structure, as found in natural languages.

Broadly, this thesis aims to contribute to the advancing of statistical models of cognition, eventually towards a statistical learning account of higher order structure. Statistical learning, alongside *formal grammars* and *connectionist models*, form three broad categories of cognitive modelling approaches.

A *formal grammar* approach defines structure by a set of production rules that transform sequences of terminal and non-terminal symbols into syntactically permissible sequences. In relation to the Chomsky Hierarchy (Chomsky, 1956, see also Appendix B for a summary of the hierarchy of formal grammars), significant parts of musical structure (specifically Western tonal harmony) can be successfully described by context-free grammars (Lerdahl & Jackendoff, 1983; Rohrmeier, 2011; Steedman, 1984), whilst natural languages are commonly considered to require a formal grammar between the levels of a Type 1 (context-sensitive) and Type 2 (context-free) grammar for parsing (Shieber, 1985). By way of reference, a statistical learning approach using a Markov model would be considered as a probabilistic regular, or finite-state, grammar (Type 3). The production rules of a context-free grammar may only replace a single non-terminal symbol with a sequence of terminal and non-terminal symbols. Since the original non-terminal symbol may also occur in the replaced sequence, context-free grammars naturally account for recursive structure. The resulting language may contain phrases where an event is structurally linked with an event indeterminately far in the future, with an arbitrarily large number of possible sequences in between. However, the production rules are assumed to be known prior to experiencing even the first utterance, inviting the question: 'where are they from?' Even if they are assumed to be innate the question is merely deferred: 'how do they arrive in an evolved mind?' By contrast, statistical models rely only on low-level, fundamental, innate processes that may be applied across domains; the rest of the structure is learned through exposure to information. Whilst easily accounting for local, transitional structure, most statistical approaches struggle to fully capture higher order phenomena such as long-term dependencies and centre embedding; in other words, phenomena that require recursion.

The final category consists of *connectionist* approaches to cognitive modelling including neural networks (originating from Hopfield, 1982), and more recently, deep belief networks (for a general review, see Bengio, 2009). Both network approaches are capable of finding structure in, and closely fitting to, complex, high dimensional data across a diverse range of domains, often defining the state of the art in various engineering, classification, and information retrieval tasks. Neural networks owe much of their power and versatility to their ability to embody non-linear functions, resulting from a highly connected, often multi-layered network of neurons, each with an activation function and weighted connections. A Hebbian learning principle (Hebb, 1949) underpins the learning mechanism, increasing the weights of connections when the activation of their neurons correlates. Superficially, their implementation matches that of a brain's complex, massively connected network of neurons and synapses, however, several serious drawbacks reduce their usefulness as approaches to cognitive modelling. Firstly, it has been argued (Cleeremans & Dienes, 2008; Rohrmeier, 2010) that a neural network is essentially a sophisticated regression model, with an extremely large number of parameters (weights) enabling a fit to any arbitrary (even apparently unstructured) data, diminishing their explanatory potential. Secondly, connectionist models are essentially black boxes (especially in the case of recurrent networks) making verification of internal processes almost impossible. On the other hand, statistical models can be extremely transparent in their application, enabling claims on the behaviour of individual components of the model to be tested and potentially falsified.

The current research develops and extends existing statistical models of cognition and perception (specifically, musical expectation), working towards models that have the potential to model and explain higher order structure. The line of research presented in this thesis can be viewed as an exploration of the limits of the capabilities of a purely statistical approach.

1.2 Scope and Domain

The present research is drawn from a number of related research areas. Fundamentally, it is situated in the field of cognitive science, in developing an understanding of the cognitive processes that account for the learning and perception of sequential, symbolic data. The present research takes a computational approach (Desain, Honing, Vanthienen & Windsor, 1998; Wiggins, 2011) in modelling cognitive processes, focussing on models that can be implemented, empirically tested, and falsified (Popper, 1934). Like many scientific models, computational models operate at a relatively high level of abstraction in comparison to the cognitive process being modelled. A computational model need not capture the low-level activity of firing neurons to adequately model a cognitive process, in the same way that an engineering model of fluid dynamics need not capture the velocities of individual water molecules in order to design an efficient water pump. The systems presented in this thesis model cognitive processes at the functional level; the purpose, input, and output of components are important, rather than the specific mechanisms and implementation that achieves these goals.

The primary task of the computational models developed in the current research will be to predict, and capture statistical structure from, events in the domain of music.

Three qualities make music an attractive subject of study in cognitive science. Firstly, musical participation is universal across all known human societies (Blacking, 1995), whilst non-functional, independent, and engaged musical participation remains unobserved in all other species. Therefore, to study the cognitive processes driving musical understanding is to study a truly unique human ability and activity. Secondly, musical languages (for example, Western tonal music) exhibit sufficient complexity, and specifically, higher order structure. Importantly, both melodic and harmonic aspects of music can, and are, learned implicitly (Cleeremans & Dienes, 2008; Rohrmeier, 2010), in much the same way that spoken natural languages are learned from infancy without formal training. Finally, music holds a methodological advantage over natural language modelling, in that referential semantics do not exist in music except in exceptional circumstances (e.g. programmatic music), or in a looser, more metaphorical sense (e.g., Larson, 2012; Zbikowski, 2002). Specifically, music cannot make propositional statements relating to the real world, while natural languages produce truth functional sentences. Whilst being forced to refer to things in the real world is problematic when implementing a computational model of language cognition, the issue can largely be ignored for music. The musical domain allows for *explanatory* models of music to be developed, and compared with existing *descriptive* models, possibly in music theory or in existing computational models. Wiggins (2007) makes a distinction between models of cognition which are explanatory and ones which are descriptive. Descriptive models describe behaviours in relation to a stimulus in terms of ‘what’ and ‘when’, whilst explanatory models work towards an account of the underlying theory behind the behaviour, in other words, ‘how’. Music, specifically western tonal harmony, provides a domain of study that clearly exhibits higher order structure; at the most basic level the tonal centre established at the start of a piece is typically departed from and returned to at the end. Furthermore, a rich vein of musicological literature (e.g., Lerdahl & Jackendoff, 1983; Riemann, 1895; Schenker, 1979) and computational models (e.g., Marsden, 2010; Rohrmeier, 2011; Steedman, 1984) are already well-established, providing a wealth of descriptive models against which an explanatory model can be validated.

The type of computational models implemented in the current research are statistical models, motivated by a body of behavioural and neurophysiological research supporting statistical learning in language and music (Jonaitis & Saffran, 2009; Pearce, Ruiz, Kapasi, Wiggins & Bhattacharya, 2010c; Saffran, Aslin & Newport, 1996; Saffran, Johnson, Aslin & Newport, 1999). Markov, or n-gram, models are the specific class of statistical models employed; given a sequence of events the probability of any event is dependent only on the preceding $n-1$ events. Such models may be naturally applied to model expectation and prediction. The next event is predicted given the preceding events, and when it arrives the extent to which it matches the prediction is assessed. The loose concepts of expectation and certainty of prediction can be quantified with the information theoretic measures of information content (MacKay, 2003) and entropy (Shannon, 1948). Indeed, information theory is key to implementing the statistical models proposed, and provides an empirical and quantifiable method for understanding the mind as, fundamentally, a processor of information (Dennett, 1991, 1996).

Probabilistic prediction, Markov modelling, and information theory are brought together by *multiple viewpoint systems* (Conklin & Witten, 1995). Multiple viewpoint systems extend Markov approaches to domains where data is fundamentally multi-dimensional, such as symbolic musical sequences. A viewpoint is a dimension, or feature, of a musical event, which may be a basic attribute of the musical surface (such as pitch or onset time) or may be abstracted from the musical surface with some function (such as pitch interval or metrical beat). At the core of multiple viewpoint modelling is a collection of techniques used to combine predictions from statistical models over different viewpoints into a single, coherent prediction. The statistical models of a multiple viewpoint system are variable order; rather than the probability distribution being conditioned on only the preceding context of $n-1$ events, they blend probability distributions from a number of differing length contexts, potentially unbounded in length.

Being essentially Markovian (or finite-context models), multiple viewpoint systems are a powerful approach to capturing local structure. Although some specialised viewpoints[1] create sequences of non-adjacent events by extracting events at pre-defined points in a sequence (e.g. the first beat of a bar), multiple viewpoint systems do not naturally lend themselves to modelling higher order structure. Firstly, the higher order structure extracted is rigid, and secondly, it is pre-defined from the basic attributes, rather than being dynamically learned. The present research makes specific contributions to the multiple viewpoint framework, before extending it to parallel viewpoint systems working at different temporal levels, with predictions from different temporal levels combined in much the same way that viewpoint predictions are combined. The proposed model is an initial implementation of a cognitive architecture posited by Wiggins (2012c) and Wiggins and Forth (2015), that aims to account for a range of human behaviours including expectation, learning higher order structure, ambiguous parsing, and creativity.

1.3 Research Aims and Contributions

This thesis aims to make contributions to the field of music cognition in providing an empirically testable computational account of statistical learning. More broadly, the

[1] The class of viewpoints known as *threaded viewpoints*, see §3.3.2.

mechanisms driving the statistical learning models are highly general, adapting to training data from any symbolic domain. Indeed, multiple viewpoint systems have been successfully applied to the musical domains of monophonic melodic prediction (Conklin & Witten, 1995; Pearce, 2005), and polyphonic melodic prediction (Whorley, 2013), as well as to phoneme sequences of natural languages (Wiggins, 2012a). Whilst the computational models of the present research are primarily trained and tested with chord sequences, they may be applied to any number of sequential symbolic domains. The contribution, therefore, may be more broadly applied to cognition in general, rather than being restricted to musical cognition.

The overarching objective of the thesis is to develop computationally implementable models of statistical learning primarily in the domain of harmony, building towards ones that may account for some degree of higher order structure. With this objective in mind, the following specific aims are identified:

1. To enhance the understanding of established statistical approaches, namely multiple viewpoint systems, by empirically testing them over novel domains, namely tonal harmonic chord sequences.

2. To develop and implement an empirically testable statistical model that has the potential to capture higher order structure as evident in tonal harmony through statistical induction.

3. To computationally test the proposed statistical model across its parameter space, measuring performance empirically with information theoretic measures, and behaviourally by its ability to identify musical structure.

The first aim is a necessary step, firstly in order to fully understand the performance and workings of multiple viewpoint systems over relatively novel domains, and secondly, so that multiple viewpoint systems may be applied as the basis to the statistical models proposed in the second and third aims.

The present research aims to contribute considerably to the field of multiple viewpoint modelling. Firstly, it is anticipated that the rigorous application of multiple viewpoint modelling techniques to a novel domain, harmonic chord sequences, may assist in validating the claim that they are a highly general statistical approach. Secondly, some relatively unexplored and underdeveloped components of multiple viewpoint systems may be explored in more detail. As discussed in §2.4, multiple viewpoint systems have a wide range of applications, and so making improvements to, and enhancing the understanding of, these models has potentially wide reaching benefits.

The statistical models developed are essentially unsupervised machine learning techniques, taking unlabelled data, learning structure through statistical induction and applying the learned knowledge to a task. More specifically, the information theoretic performance metrics employed evaluate a model's ability to compress data efficiently, opening the possibility of contributions to the field of data compression. However, it is worth noting that the purpose of the research is to develop models that are informative of cognitive processes, and not necessarily to develop a model that outperforms the state of the art in an engineering task.

Finally, the research aims to make indirect contributions to the field of computational creativity, in developing systems that have the ability to learn structure, knowledge, and representations in an automated manner. Computational creativity is the development of systems that exhibit behaviour that would be considered creative to an unbiased observer (Colton & Wiggins, 2012). A fundamental distinction between a truly creative computational system, and a system which merely outputs novel artefacts, is in its ability to learn independently (Boden, 2003; Wiggins, 2012c). A system that simply regurgitates, for example, poetry through some pre-defined combinatorial rule set may produce interesting works, but the behaviour of the system could not be considered particularly creative as there is no ability to self-reflect, to understand emotion or semantics, and the rule set that produces the interesting works is defined by a human. Furthermore, for computational creativity to be exhibited in the musical domain, higher order structure, as well as local structure, must be learned. A creative system with the ability to automatically learn complex structure and representations has the potential to exhibit the highest form of creativity, *transformational creativity* (Boden, 2003). Transformational creativity requires the transformation of a conceptual space, or high-level representation (Ritchie, 2006), in contrast to mere exploratory creativity, which searches for novel artefacts within a defined conceptual space. The contribution of the present research extends only to learning structure, and not to the task of generating novel works.

1.4 Thesis Outline

An outline of the thesis is presented in four parts. Part I is an exposition of the related literature and theoretical foundations for the current research. Part II presents a body of work relating to multiple viewpoint systems (Conklin & Witten, 1995); proposing and empirically testing a number of improvements, and providing a deeper understanding in their operation across various domains. The primary aim of Part III is to develop a preliminary, computationally testable, implementation of a cognitive architecture proposed by Wiggins and Forth (2015), with a focus on statistical learning, information theoretic prediction, and higher order structure. Finally, Part IV reflects on the research presented in this thesis, assesses potential shortcomings, and outlines future directions.

Part I: Overture

Chapter 2 establishes the position of the research; making the argument that expectation-driven statistical learning accounts for human musical behaviour. Several relevant areas of research are reviewed, including the cognition and perception of tonal harmony, and computational models of tonal harmony. The chapter then provides a comprehensive review for the development and applications of multiple viewpoint systems.

Chapter 3 provides a detailed theoretical description of multiple viewpoint systems, collating the works of Conklin and Witten (1995), Pearce (2005), and Whorley (2013). The work is presented at a level of description sufficiently detailed to reproduce a full, computational implementation. Multiple viewpoint systems are an established representational framework and associated probabilistic (principally Markovian) modelling scheme developed to capture statistical structure of multidimensional, sequential, symbolic data. Originally conceived for musical melodic prediction (Conklin, 1990), multiple viewpoint systems are highly generalisable, and can be applied to modelling symbolic data from a wide range of domains. They are the primary modelling technique of the thesis, studied in depth in Part II, and providing the underlying statistical models for Part III.

Part II: Developments in Multiple Viewpoint Systems

The corpora and representational viewpoints used in the current research are given in Chapter 4. The main contribution of the chapter is in providing a viewpoint representation scheme for chord symbol sequences, as would be found in music notation on a lead sheet. In a similar manner to Chapter 3, a level of description is provided that is sufficient to fully implement the models used throughout this thesis. Some methodological and ontological issues relating to the musical surface, and modelling temporal structure with multiple viewpoint systems are discussed.

Chapter 5 turns the attention of the thesis to empirical testing of multiple viewpoint systems, namely, addressing issues concerning the prediction of two or more attributes (or features) of a musical event. A novel method is presented, whereby attributes are merged into a single attribute, potentially taking advantage of strong correlations between the attributes. This particular representational approach minimises the information lost when attributes are matched individually when counting in statistical models, whilst retaining the full descriptive power of the multiple viewpoint framework. In addition, the chapter empirically compares various *smoothing techniques*,[2] extending the findings of Pearce and Wiggins (2004) from the monophonic melodic domain, to the harmonic (chord sequence) domain.

Chapter 6 proposes and tests a new technique for improving the predictions of a class of viewpoints known as derived viewpoints.[3] The technique is tested for both individual viewpoints, and whole multiple viewpoint systems. The results invite a discussion on the implications of increased computational cost for improved model performance, which has implications in the latter stages of the thesis, and computational models of cognitive processes in general.

Chapter 7 presents an in-depth assessment of the viewpoint selection algorithm, an algorithm used to automatically build locally optimal multiple viewpoint systems. The assessment explores the search's state space by using random initialisation points in order to assess whether the local optima found by the original algorithm are representative of other locally optimal solutions.

Finally, Chapter 8 compares the predictive performance of absolute viewpoints, such as pitch and root, with relative viewpoints, such as pitch and root intervals. The consensus of the multiple viewpoint literature (Pearce & Wiggins, 2012) is that relative viewpoints produce more information theoretically efficient models than absolute viewpoints, motivating their primary status in cognitive representations. This chapter tests these claims in detail across a range of datasets and domains.

Part III: Statistical Learning and Higher Order Structure

The Information Dynamics of Thinking (IDyOT) model is introduced in Chapter 9, summarising the theoretical and philosophical foundations of the model from the works of Wiggins (2012c), Wiggins and Forth (2015), and Forth, Agres, Purver and Wiggins (2016). IDyOT is a cognitive architecture driven by statistical and information theoretic processes, capable of accounting for key human behaviours such as consciousness, learning complex structure, expectation, cognitive representations, and creativity. As a relatively young model, the current state of the art in IDyOT is considerably less developed than that of multiple viewpoint systems; at the time of writing no full im-

[2] A collection of methods for improving predictions in n-gram or Markov models.

[3] Derived viewpoints are a class of viewpoints that do not model an attribute directly, but are able to make more general predictions by abstracting information away from the basic attribute, for example a viewpoint representing the pitch interval may be used to model pitch itself. A full description of viewpoint classes is given in §3.3.2.

plementation of IDyOT exists. The chapter, therefore, aims to provide a high level description of the core processes of IDyOT, with a view to developing a computational implementation in subsequent chapters.

Chapter 10 presents an exploratory implementation of aspects of the IDyOT cognitive architecture, focussing on the components associated with the tasks of chunking and prediction. A degree of higher order structure is found by chunking the surface layer, forming chunks on an upper layer which in turn inform the prediction of subsequent symbols on the surface layer. The implementation is computationally tested across its free parameters, with each set of parameters allowing the behaviour of specific components of the architecture to be empirically observed and tested.

Some shortcomings of the IDyOT implementation presented in Chapter 10 are assessed and partially addressed in Chapter 11. Although the ultimate goal of IDyOT is a cognitive architecture capable of automatically learning any domain purely through statistical exposure, this updated implementation takes the approach of providing some domain-specific knowledge to a few components in order to observe the statistical learning processes of others. The higher order representation in the updated IDyOT implementation is given some musicological knowledge related to tonal harmony, using it to represent chunk sequences in relation to a tonal centre. After an empirical exploration of the free parameter space, the behaviour of the model is assessed in detail: namely, its potential to produce musicologically meaningful parsings, and its ability to produce segmentations of jazz chord sequences that correlate with other reliable segmentations.

Part IV: Coda

Chapter 12 discusses and draws conclusions from both Parts II, and III of the thesis. Potential avenues for future research are presented, including further developments in implementing IDyOT, re-assessing information theoretic evaluation of computational models of cognition, and automatically learning representation schemes such as viewpoints.

Chapter 2

Literature Review and Related Research

2.1 Overview

The research and literature providing the foundation to this thesis falls into three main groups. §2.2 builds an understanding of perceptual and cognitive processes in music, principally those relating to expectation, statistical learning, and tonal harmony. Computational models of tonal harmony are reviewed in §2.3, with a distinction drawn between *learned* and *non-learned* models. Finally, a comprehensive review of the field of multiple viewpoint modelling is conducted in §2.4. A contextualisation of the present research closes the chapter (§2.5).

2.2 Music Perception and Cognition

With the overall goal of developing computational models for music cognition (specifically in tonal harmony and higher order structure), a number of key research areas relating to the perception and cognition of music are reviewed in the following section. First, motivations are established for modelling music cognition within the context of expectation, and then discussed with reference to the works of Meyer (1956), Narmour (1990), and Huron (2006). This forms the foundation of a statistical learning approach to cognition, and a review of the salient findings related to statistical learning of music. Finally, a summary of research on expectation and statistical learning relating to tonal harmony is provided.

2.2.1 The Role of Expectation

> *"The task of the mind is to produce future, as the poet Paul Valéry once put it. A mind is fundamentally an anticipator, an expectation-generator."*
> *Kinds of Minds*, Dennett (1996, p. 57)

Expectation is heralded as the primary function of the mind in Daniel Dennett's philosophical theories on conciousness (Dennett, 1991, 1996). Undeniably, as an evolutionary mechanism, the ability to anticipate events, predict outcomes, and identify uncertain situations is advantageous for survival. Whether music developed as a result of exaptation, functioning as Pinker's (1997) *'auditory cheesecake'*, or as a result of social evolution and adaptation (see Cross, 2001 for a review), the fact remains that the cognitive mechanisms underpinning prediction and expectation are core components of the cognition, perception, and production of music. Indeed, a sizeable battery of empirical evidence suggests musical expectation plays a significant role in musical perception (Cuddy & Lunney, 1995; Pearce & Wiggins, 2006; Schellenberg, 1996), musical memory (Schmuckler, 1997), emotional responses to music (Egermann, Pearce, Wiggins & McAdams, 2013; Steinbeis, Koelsch & Sloboda, 2006), the production of music (Schmuckler, 1990; Thompson, Cuddy & Plaus, 1997), the perception of segment boundaries (Pearce, Mullensiefen & Wiggins, 2010b), and neurophysiological responses to music (Gebauer, Kringelbach & Vuust, 2012; Koelsch, Busch, Jentschke & Rohrmeier, 2016; Koelsch, Kilches, Steinbeis & Schelinski, 2008; Pearce et al., 2010c; Steinbeis et al., 2006).

Reinforcing this line of thinking, Meyer (1956) establishes expectation as the core mechanism by which meaning and emotions are elicited in music perception. It is argued that expectation in music is governed by (statistical) musical structure, characterised by three categories of expectation violation. Firstly, expectations established by a preceding context can be delayed, secondly, uncertainty can be evoked when a preceding context generates no strong expectations, and finally, a consequent event or musical pattern may be unexpected given its context, evoking surprise. Perhaps the most significant aspect of Meyer's theory is the absence of referential semantics in an account of musical meaning; musical structure and syntax themselves are sufficient components to account for such a phenomenon.

The *Implication-Realisation (IR)* theory of Narmour (1990) furthers this approach with a more complex, and importantly, quantifiably specified model of melodic prediction. Distinct *bottom-up* (innate, universal), and *top-down* (learned from musical experience) systems are posited to account for musical expectation. Melodic expectation is defined in terms of sequential intervals; *implicative* intervals (intervals that do not provide melodic closure) generate expectations for certain *realised* intervals acting as resolution. The bottom-up system is built on Gestalt principles defined by a set 12 melodic structures. Briefly, the principles behind these structures can be summarised such that small implicative intervals imply expectation of small realised intervals in the same direction, whilst large intervals imply a smaller interval in the opposite direction. The strength of the expectations elicited by implicative intervals can be graded according to the degree to which they comply with these principles. Owing to the precise definitions of the melodic structures in the theory, the IR model lends itself naturally to being quantified and empirically tested as a psychological and perceptual theory of music. Krumhansl (1995) formulates the bottom-up aspects of the IR model quantitatively as a set of symbolic rules, which can be further simplified to a two-factor model (Schellenberg, 1996, 1997) when used as an empirical model in predicting listener's responses in melodic continuation probe tone studies (Cuddy & Lunney, 1995; Schellenberg, 1996). Although the bottom-up components are considered innate in Narmour's (1990) original theory, Pearce and Wiggins (2006) show that they can largely be accounted for with a statistical learning model trained on folk songs, ballads, and chorale melodies when tasked with predicting listener's melodic expectations in a variety of settings (Cuddy & Lunney, 1995; Manzara, Witten & James, 1992; Schellenberg, 1996).

Huron (2006) presents a theory of musical expectation with more statistically explicit underpinnings. Again, an evolutionary rationale is given for developing advanced cognitive processes to handle auditory expectation. Notably, Huron identifies a need for valenced emotional responses as reward and punishment systems to non-fatal events. An evolutionary account of expectation consisting only of fatal punishment systems is deeply flawed; it is difficult for an organism to learn from an experience if it is dead. Huron argues this gives rise to associating positive and negative emotions with correctly and incorrectly predicted events, as well as with certain and uncertain predictions.[1] These specific and complex emotional responses tied to expectations form the foundation of the sophisticated cognitive processes capable of accounting for musical behaviour.

The cognitive process of expectation is summarised by Huron (2006) with five stages in the *ITPRA* theory as follows. Before an event the *Imaginative response* imagines and evaluates potential outcomes, and the *Tension response* adjusts arousal levels according to the predictability and importance of the imminent event. After the event, the *Prediction response* assesses the extent to which predictions made before the event were correct, the *Reactive response* is a fast, automatic response reacting to the event, and the *Appraisal response* is a slower, more explicitly conscious assessment of the event and predictions, adjusting for future events with positive or negative reinforcement related

[1] *Contrastive valence* is proposed as an account of experiencing positive emotions with unexpected, but positive, events; the unexpected event heightens the limbic response which retrospectively turns out to be positive in the reaction and appraisal phases.

to the outcome. The underpinning mechanisms behind the theory (especially auditory expectations) are, therefore, fundamentally statistical.

2.2.2 Statistical Learning

Statistical learning theories find their origins in natural language acquisition, hypothesising that the necessary cognitive processes can be acquired through statistical induction alone (Manning & Schütze, 1999; Rebuschat & Williams, 2012). The theory contrasts sharply with a Chomskian (Chomsky, 1957) approach to natural language, where a *poverty of stimulus* argument motivates the notion that complex recursive language structures are considered innate, even quasi-platonic (Wiggins, Mullensiefen & Pearce, 2010). However, such approaches usually place a binary membership on sentences as grammatical or un-grammatical according to a rule set.[2] A statistical account lends itself to a more probabilistic approach, which better accounts for everyday applications of language where fuzzy categorisations of utterances in terms of both grammatical correctness and semantics are evident. The critical behavioural study of Saffran et al. (1996) shows that 8-month old infants use statistical cues in the form of transitional probabilities when identifying word boundaries of nonsense syllables in an artificial grammar. The empirical evidence of the study considerably weakens the poverty of stimulus argument, and established the research area as a key field in cognitive science and natural language processing, prompting statistical learning studies for other cognitive tasks (see Rebuschat & Williams, 2012).

Given the drive for overarching cognitive theories capable of describing behaviour and phenomena in multiple domains, the application of statistical learning theories in music cognition is perfectly natural. To date, empirical evidence has been established for statistical learning accounts of a wide range of musical behaviour. The seminal work of Krumhansl (1990) correlates melodic pitch cognition with basic statistical structure in large corpora of Western tonal music. An important methodological contribution of this line of research is the *probe-tone paradigm* (Krumhansl & Shepard, 1979) where participants give explicit goodness-of-fit, or continuation ratings for tones proceeding a particular melodic context. By keeping fixed the melodic context and varying the probe tone pitches, a pitch class profile across all pitch classes gives an overall picture of the relation between specific pitch classes and the controlled context. The early probe tone studies (Krumhansl & Kessler, 1982; Krumhansl & Shepard, 1979) established the concept of a tonal hierarchy whereby goodness-of-fit ratings over scale degrees (pitch

[2] Probabilistic context-free grammars (Manning & Schütze, 1999, pp. 381-405) being the notable exception.

relative to an induced tonal centre) varies systematically. Krumhansl (1990, ch. 3) makes the argument that these tonal hierarchies can be accounted for by simple frequency counts of scale degrees in Western tonal music corpora (Knopoff & Hutchinson, 1983; Youngblood, 1958), with higher correlations found for major, rather than minor, keys, and a multiple regression analysis revealing that a frequency-based model subsumes a tonal consonance model in accounting for the observed tonal hierarchies.

The underlying statistical models in the initial exploration of Krumhansl (1990) are relatively basic: simple unigram distributions of pitch and scale degree counts. However, musical structure clearly exceeds simple frequency counts with pitch, chord, and temporal onsets dependant on their immediately preceding context, their position in tonal, harmonic, or rhythmic hierarchies, as well further top-down influences associated with musical cultures, styles, and genres. The first of these factors, the context, can be represented by statistical models such as first-order Markov models, which track transitional probabilities between events. Higher order Markov models take into account longer contexts, and more advanced models still, variable order Markov models (Begleiter, El-Yaniv & Yona, 2004), take into account multiple contexts of varying lengths. The Information Dynamics of Music (IDyOM) model (Pearce, 2005; Pearce & Wiggins, 2012) is an advanced, variable order, multidimensional statistical model, capable of making considerably more accurate and specific predictions than the simple frequency distributions utilised by Krumhansl (1990). Trained with large melodic corpora such as folk songs, ballads, and chorale melodies, a series of behavioural studies empirically show IDyOM is a powerful model of melodic expectations (Pearce et al., 2010c; Pearce & Wiggins, 2006), segmentation (Pearce et al., 2010b), and memory (Agres, Abdallah & Pearce, 2017). A more in-depth review of cognitive and perceptual studies using IDyOM is given later in this chapter (§2.4.3), with a full description of the statistical model itself given in Chapter 3.

Taken alone, the correlation between statistical patterns evident in large corpora and empirical behavioural results implies only that statistical models have sufficient descriptive power to account for said empirical behavioural results, not necessarily that they are the underlying causes. An alternative explanation of the results might place the chain of cause and effect in the opposite direction; innate cognitive schema relating to pitch expectancy are implicitly understood by composers, and so are reflected in the statistical structure of their compositions. In an effort to understand this complex process in more detail two methodologies have been adopted using unfamiliar musical styles, and artificial grammars.

In the first category, cross-cultural studies are able to make comparisons between participants of different cultures that have not yet been exposed to various musical styles, suggesting any similarities in results can be accounted for with innate factors, and differences with learned (possibly statistically) factors. Krumhansl et al. (2000) present a behavioural and statistical study of melodic expectancy in North Sami yoiks[3] by three groups possessing varying familiarity with the style (indigenous Sami people familiar with the yoiks, Finnish music students who had learned some yoiks, and western musician with no exposure to the Sami music). The probe tone study found broadly similar responses between the three groups relating to completion tones, confirmed by an analysis of variance returning adjusted r^2 values of between 0.71 and 0.79 for an IR model (Narmour, 1990) simplified to five bottom-up components (Krumhansl, 1995). However, an analysis of individual components across groups revealed relatively lower correlations for the *intervallic difference, registral return,* and *proximity* components in the Sami and Finnish groups compared to the Western participants. Additionally, Western participants were predictably found to be influenced predominantly by Western schematic knowledge (Krumhansl, 1995). Two self-organising map neural networks (Kohonen, 1997) trained separately with Finnish melodies and yoiks were found to correlate well with their corresponding groups. A further study (Eerola, 2004) found a statistical model capturing pitches and pitch intervals accounted for the behavioural data better than simplified IR models and auditory memory models. Further cross-cultural studies on melodic expectation in general conform to the notion that the main patterns of expectation are broadly similar between familiar and unfamiliar participant groups, however, a difference is found in the intricacies of more detailed expectation patterns, which can be accounted for with statistical models of varying complexities (Castellano, Bharucha & Krumhansl, 1984; Eerola, 2003; Krumhansl, Louhivuori, Toiviainen, Järvinen & Eerola, 1999).

Cross-cultural approaches are increasingly problematic in the modern world as the dissemination of musical styles becomes easier, thereby confounding the notion of musical familiarity between subject groups. An alternative approach guaranteeing unfamiliar stimuli is the artificial grammar paradigm, where a novel grammar is used to systematically construct stimuli ensuring both an underlying musical structure, and unfamiliarity for participants. One such artificial grammar system based on the 13-step Bohlen-Pierce scale (Mathews, Pierce, Reeves & Roberts, 1988) is constructed and used for a series of statistical learning behavioural experiments summarised by Loui (2011). A probe-tone study (Loui, Wessel & Hudson Kam, 2010) tested participants before and after a training phase consisting of prolonged and repeated exposure to melodies in the artificial grammar. Ratings were found to correlate significantly better with an exposure profile (relating to the frequency of pitches in the training phase) after the

[3] A traditional song with chanting qualities associated with the cultures of various Nordic countries and the Kola peninsular.

training phase, compared to before, suggesting general statistical learning independent of musical systems are capable of accounting for melodic cognition. Further experiments (Loui & Wessel, 2008; Loui et al., 2010) used a two-alternate forced-choice paradigm to investigate the extent to which melodies can be identified that do or do not comply with an underlying harmonic grammar in the Bohlen-Pierce system after participants have experienced a training phase which varies the number of different melodies and the number of repeated exposures. Taken together (Loui, 2011), results show that whilst exact recognition is high when a small number of melodies are repeated a large number of times, generalisation (the ability to identify melodies that have not been heard, but comply with the harmonic grammar) is poor. Conversely, generalisation improves when the number of different melodies is increased, to the point where it exceeds recognition accuracy when 400 different melodies are exposed only once each. Loui et al. (2010) suggest both musicians and non-musicians posses rapid implicit learning mechanisms capable of generalising knowledge of the underlying harmonic grammars not immediately available in the melodic transitions in the training stimuli. Other melodic studies using artificial grammars based on finite-state machines show that participants with contrasting musical cultural backgrounds learn melodic structure consistently as shown by forced-choice familiarity ratings, and binary confidence ratings (Rohrmeier, 2010, ch. 2). However, recognition performance has been found to significantly decrease in further experiments with stimuli that deliberately contravened Narmour's (1990) IR model (Rohrmeier, 2010, ch. 3).

Overall, empirical evidence suggests statistical learning may account for a wide range of musical phenomena; in addition to those discussed above these include first-order harmonic transitions (Jonaitis & Saffran, 2009), context-free harmonic grammars (Rohrmeier & Cross, 2009), online (within test) learning (Rohrmeier & Cross, 2014; Rohrmeier, Rebuschat & Cross, 2011), perceptual melodic similarity (Eerola, Jäärvinen, Louhivuori & Toiviainen, 2001), absolute pitch judgements (Miyazaki, 1989; Simpson & Huron, 1994), and grouping in melodies (Saffran et al., 1999) and timbre sequences (Tillmann & McAdams, 2004).

2.2.3 Tonal Harmony

There is an important distinction to be considered when modelling tonal harmony[4] relating to the perspective from which it is being studied. Tonal harmony can be viewed from a purely musicological perspective, where structure is inherent at the representational

[4] These distinctions apply to the study of music in general, however the following specifically relates to tonal harmony as one of the core target domains of the thesis.

level of musical scores. Given that Western tonal music typically starts in a home key, modulates to one or more keys in a stylistically appropriate manner, before returning to the home key, a level of grammar with sufficient descriptive power to account for the variety of long-term dependencies evident seems necessary. The level of structural complexity inherent in tonal harmony itself can, therefore, be argued to be equivalent to a language generated from a context-free grammar (Rohrmeier, 2011). This is perfectly plausible given that a composer has no theoretical bound on revisions to a score, and is not temporally limited in the way that they experience the score (in contrast to a listener who experiences music in real time).

When considering some of the best established models of tonal harmony (Lerdahl & Jackendoff, 1983; Schenker, 1979) this argument appears to be borne out; recursive processes are the most efficient way to account for the tonal harmonic structure observed in the music itself. Conversely, a model might aim to describe the cognitive processes accounting for the learning, understanding, and perception of tonal harmony. In this case, the objective of the modelling process is not to construct a *de facto* account of tonal harmonic structure, but to further the understanding of the processes in the mind of the listener as they experience music. In doing so, considerable constraints must be considered; at the most basic level music is (usually) experienced in real time. Perhaps more problematic are the host of perceptual, cognitive, and psychoacoustic constraints and phenomena which are only partly understood through the neurophysiological, behavioural, and psychological empirical research that has been carried out to date. With the current research aiming at contributing computational models for the cognition of tonal harmony, the following section reviews this area of research.

At the most basic level of harmonic perception, there is strong perceptual evidence supporting the concept of harmonic relatedness, which has been established theoretically with mathematical structures such as *tonnetz* (Longuet-Higgins, 1978) and *spiral arrays* (Chew, 2002). Perceptual relatedness quantified with explicit participant ratings shows that closely related tonal contexts play a role in establishing perceived closeness (Krumhansl, Bharucha & Kessler, 1982b), with multidimensional scaling and clustering analyses grouping the tonic, dominant and subdominant harmonic functions. Perceived relatedness between chords is found to be closer with a related or matching tonal centre in comparison to no tonal context (Bharucha & Krumhansl, 1983), or distant tonal centres (Krumhansl, Bharucha & Castellano, 1982a). Expectancy is temporally directional, and as such can be viewed as a more specific concept than perceptual relatedness. Bharucha and Stoeckig (1987) use a priming paradigm (measuring processing time when judging out of tune pitches in chords) to show that mere frequency repetition between prime-target pairs does not sufficiently account for priming effects. Instead, harmonically related chords induced strong priming effects, suggesting activation of processes at the cognitive, as opposed to perceptual, level. A series of probe-chord experiments (Schmuckler, 1989) with increasingly ecological validity show that explicit expectation ratings broadly correlate with Piston's (1948) table of root progressions, although considerable variation is found in all but the most likely progressions. Taken together, these studies indicate that basic musicological concepts such as tonal harmonic distance and common harmonic progressions are accounted for in a listener's perceptual and cognitive processes.

The body of statistical learning research reviewed in §2.2.2 is predominantly in the melodic domain. However, there is substantial evidence to suggest that statistical learning is also applicable to the harmonic domain. Statistical learning of chord transition probabilities is investigated with an artificial grammar paradigm in Jonaitis and Saffran (2009). Two artificial Markovian grammars of chords in the Phrygian mode are counterbalanced so that chord transition probabilities are reversed between grammars: $p(A|B) \propto p(B|A)$ after normalising distributions and adjusting for a termination state designed to end sequences from both grammars on the tonic. Crucially, frequency counts of individual chords is consistent between the two systems; only the transitional probabilities differ. A test set of 60 chord sequences include 30 'correct' (15 from each system) items, and 30 'error' items (with one, two, or three chord transitions that violate one of the systems). After an implicit training phase exposing subjects to 100 chord sequences from one of the grammars, subjects gave explicit similarity ratings on a seven-point scale comparing test chord sequences against the exposure corpus. An analysis of variance showed that whilst subjects were able to differentiate between test sequences generated by the other grammar, they were unable to identify 'error' items generated from either system. This result held when the test items only contained items from the grammar used in the exposure phase. However, with increased exposure and a one day time interval between training phases, participants were able to identify 'error' items, suggesting a sufficiently fine-grained understanding of the artificial grammar required memory consolidation.

Beyond mere first-order transitions, Rohrmeier and Cross (2009) finds empirical evidence that statistical exposure may account for hierarchical structure at the level of context-free grammars. An artificial grammar based on the octatonic system consisting of three terminal symbols, three non-terminal symbols, and three production rules capable of producing recursive, centre-embedded (on the first level only) structure is defined. With an enforced limit of three hierarchical layers, 18 abstract structures are produced, 10 used for training and 8 for testing, from which individual surface exemplar chord sequences are produced. In the testing phase, new familiar items were produced from the abstract training structures, together with new unfamiliar items from the abstract testing structures, and also ungrammatical items which violated the grammar from a unigram distribution. In testing, participants performed significantly above chance at identifying the most familiar of a pair of test items, with confidence ratings indicating explicit awareness of correct and incorrect judgements. These results hold for a further experiment with a more complex grammar capable of producing centre-embeddings on all hierarchical layers. In support of these findings, it has been shown that non-local structure has a significant impact on the perception of tonal harmony in the context of memory tasks (Farbood, 2010) and probe cadence perception (Woolhouse, Cross & Horton, 2006, 2016).

The behavioural findings discussed above are broadly supported by neurophysiological research concerning expectation and harmony. Electroencephalogram (EEG) studies show that unexpected harmonic events produce a significant early right anterior negativity (ERAN) peaking in the region of 180-230ms after the event (depending on musical experience), and a late bilateral frontal negativity 500-550ms (Koelsch et al., 2008; Steinbeis et al., 2006). These are found to correlate with higher self-assessed tension and emotion ratings, as well as increased electrodermal activity in the form of a skin conductance response, with a peak found 2.5 seconds after the event. The fundamental patterns in the findings are found to be broadly similar for musicians and non-musicians, and hold for both artificially produced (Steinbeis et al., 2006), and naturally performed (Koelsch et al., 2008) stimuli. Extending the above findings to unexpected harmonic events that violate a (non-local) hierarchical structure, Koelsch, Rohrmeier, Torrecuso and Jentschke (2013) find early frontal negativity at 150ms and a later negativity around 500-850ms after the chord onset. These negativities are reminiscent of the two significant negativities found for locally unexpected harmonies (Koelsch et al., 2008; Steinbeis et al., 2006), although the first is now bilateral. The stimuli used by Koelsch et al. (2013) are short 10 second Bach chorale extracts carefully modified to violate tonal harmonic long-term dependencies by ending in the wrong key, but without violating any local chord-to-chord transitions. Overall, the study provides empirical evidence that human cognition is capable of processing hierarchical structure consisting of non-local dependencies. However, bearing in mind the short stimuli length (10 seconds) and that listeners memory, and harmonic expectancies are found to decay after 10-12 seconds (Farbood, 2010; Woolhouse et al., 2016), it seems unlikely that these findings would hold for longer pieces of music.

2.3 Computational Approaches to Modelling Tonal Harmony

The application of computational models to tonal harmony has proved fruitful in the fields of artificial intelligence, machine learning, music cognition, and computational musicology, with considerable depth exhibited in the wide variety of approaches. The following section summarises the key computational models of tonal harmony, broadly dividing them into two categories: 'learned' and 'non-learned.' Learned models of tonal harmony typically infer structure from a training set through statistical and probabilistic methods, whilst minimizing syntactic, semantic, and, potentially, representational assumptions about the target data. On the other hand, non-learned models usually take the form of (non-statistical) rule-based systems or formal grammars, with structure inferred through procedural or grammar production rules.

2.3.1 Non-learned Models

Winograd (1968) presents one of the earliest computational approaches to music analysis in general; applying a systematic grammar previously only used in linguistics to parsing tonal harmony. The work can be viewed as a formalisation of the tonal harmonic theory of Forte (1962) inspired by the formal grammars of Chomsky (1957). Potential parallels are drawn between formal grammars parsing natural language, and grammatical structure in tonal harmony, with the application of production rules revealing the syntactic structure of a chord sequence. The system is presented in the form of a systematic grammar, comprised of production rules categorised into a hierarchy of five 'ranks': composition, tonality, chord group, chord, and note. Parsing is conducted from right-to-left to reduce the branching factor, usually finding a set of syntactically valid readings for a piece. This set is systematically ordered preferring both simple (in terms of tree depth), and functionally meaningful (defined by a set of chord function rules) parsings. Testing the grammar on Schubert songs and Bach chorales proves largely successful, although unusual cadential progressions are difficult to identify owing to the lack of voice-leading knowledge in the system. Additionally, the right-to-left parsing (reading music backwards) severely limits the system as a cognitive model for tonal harmony.

Other formal grammar approaches include Steedman (1984): defining a generative grammar for 12-bar blues chord sequences. It is argued that only six rewrite rules (in addition to an initial rule defining the input sequence) forming a context-free grammar are required to generate all possible 12-bar blues chord sequences without potentially generating invalid blues sequences. The rewrite rules are derived from functional harmonic and jazz substitution principles, and the system is tested by convincingly parsing a number of blues forms from a pedagogical source. The work has been extended by Chemillier (2004) to identify and precompile cadential cadences in a real-time improvisation task. A more general, context-free grammar for tonal harmony is presented by Rohrmeier (2011) as an explicit, computationally implementable formalisation of Lerdahl and Jackendoff's (1983) Generative Theory of Tonal Music (GTTM) *prolongation reduction* principles. A Riemmannian (Riemann, 1895) functional harmonic approach is taken in defining a four-level hierarchy for production rules of *phrase level, functional level, scale-degree level*, and *surface level.* A common criticism of purely grammar-based approaches to modelling tonal harmony is that whilst they are able to account for multiple valid analyses of a piece, they are usually unable to identify 'good' from 'bad' parsings according to a meaningful metric.[5] Granroth-Wilding and Steedman (2014) addresses this issue by pairing a combinatorial categorical grammar (Granroth-Wilding, 2013; Steedman, 2000) for jazz chord sequences with a statistical learning method supervised with a set of annotated chord sequences. Adapting a probabilistic context-free grammar method, a statistical method is used to estimate the probability distributions for internal generative nodes of the parsing tree, and hence, estimate the probability of a whole tree. In a comparison against expert, hand-parsed jazz chord sequences, this method is found to outperform a baseline Hidden Markov Model (HMM).

Other non-learned models for tonal harmony in the literature do not take a strict formal grammar approach, but instead use rule-based or logic systems. Ulrich (1977) provides an early example, defining a set of rules for functional harmonic analysis underpinning a larger system for automatic melodic jazz improvisation. Rules assigning chords to functional labels (*dominant, subdominant, tonic,* and *transition*) are established and used in a left-to-right parsing which successively groups adjacent segments by tonal centre, starting from a point where each chord is in its own individual segment. The system proves to be successful in parsing simple jazz chord sequences, although the improvisation itself, building improvisation from juxtapositions of motifs, is reported to be limited. A similar approach is taken by Maxwell (1992) when producing functional analyses of Bach chorales. A large set of production rules aims to assign functions to harmonically significant vertically aligned pitches by assigning function labels to chords in relation to tonal centres. A balance between finding tonal centres which are a close fit to chord functions, and minimizing modulations to new tonal centres is made in a left-to-right parsing mechanism. However, arguably the system as an explanatory and general model suffers from being overly convoluted, with a large number of highly specific production rules which often rely on arbitrary numerical values to explain unusual

[5] Winograd (1968) achieves this to an extent by preferring simpler parsings, although it is debatable whether this will correlate strongly with musicologically meaningful parsings.

harmonic sequences.

Notably, both of the above approaches are non-hierarchical, with functional chord labels relating to tonal centres inhabiting a single level. These can be contrasted with Pachet (2000), which presents a hierarchical rule-based system for jazz chord sequences. The analysis identifies *shapes* at both low levels (e.g. *turnarounds, two-fives, two-five-ones*[6]) and high levels (e.g. *AABB*). The system is applied to recognising blues structures, and assessed with a musicological analysis of *'Solar'* by Miles Davis. Handling harmony at the note, rather than chord level, Ebcïoğlu (1986) presents an expert system for harmonising Bach chorales. A very large set of 350 production rules, constraints, and heuristics are defined in first-order predicate form from a knowledge base compiled from an expert analysis of harmony treatise and the chorales themselves (Riemenschneider, 1941). Although the system produces stylistically appealing harmonisations, the large number of production rules, approaching the size of the knowledge base itself,[7] calls into question the generality of the model. Many rules are simply used to handle exceptions found in the knowledge base not handled by the more common rules.

Marsden (2010) addresses another form of musical analysis; namely a computational implementation for automating *Schenkerian* analysis (Schenker, 1979). Schenkerian analysis, in contrast to the analyses discussed above, is a reductional analysis which aims to reduce tonal music to structurally significant notes forming a strictly defined hierarchy. A core concept is the notion of the *Urlinie*, a melodic line resulting from a reduction spanning a piece of tonal music descending from the third, fifth or eighth (octave) scale degree to the tonic. Schenkerian reduction lends itself naturally to explicit, implementable, rule-based approaches to analysis, inspiring a number of computational systems capable of Schenkerian analysis to varying degrees of success (Frankel, Rosenschein & Smoliar, 1976; Kassler, 1975, 1988; Kirlin, 2014; Mavromatis & Brown, 2004). Marsden's (2010) approach is motivated by the need to build structural representation schemes of music that are constructive, derivable, meaningful, decomposable, hierarchical, and generative (Marsden, 2005). Schenkerian reduction is formalised by defining a set of *atomic elaborations* serving as common prolongation patterns (for example, *consonant skip, appoggiatura, anticipation, suspension, repetition, interruption* etc). A chart-parser[8] algorithm stores a set of partial solutions and uses a set of heuristics to find syntactically valid analyses. Linear regression is used to define a 'goodness' metric from 9 out of 14 candidate features with an analysis using six annotated Mozart Piano Sonata themes.

[6] A turnaround is a idiomatic transnational chord sequence between sections, and *two-five* and *two-five-one* figures both relate to their functional labels of predominant-dominant-tonic (*ii-V-I* and predominant-dominant *ii-V*.)

[7] By way of counterexample, one could imagine a somewhat perverse set of production rules each of which simply harmonises a copy of a chorale in the knowledge base.

[8] The Cocke–Younger–Kasami (CKY) algorithm, see Jurafsky and Martin (2009, pp. 470-477).

Using the goodness metric to rank candidate analyses from the chart-parser and selecting the highest ranked, accurate (79-98%) and convincing analyses of five of the Mozart themes are found. The notable drawbacks of the system are the prohibitive time, up to $O(n^5)$, and space, up to $O(n^4)$, complexities. Furthermore, the small number of pieces are used both for evaluation and training the goodness metric, potentially resulting in overfitting and poor generalisation.

2.3.2 Learned Models

'Learned' approaches to modelling harmony usually take the form of statistical models, such as Markov or n-gram models, or probabilistic graphical models. An early example of a Markovian approach to modelling harmony is given by Ponsford, Wiggins and Mellish (1999); finding statistical structure in, and generating, baroque dance music. 3- and 4-gram models sample vertically aligned pitch sets every quaver (eighth note), which is found to be the shortest harmonic duration in the corpus. Non-harmonic pitch sets resulting from passing notes or suspensions are deemed to be infrequent enough to be statistically insignificant. Unusually, piece, phrase, and bar line structure is explicitly encoded as symbols in the chord sequences. This creates potentially unusual situations whereby symbols occurring simultaneously, for example a bar line and a chord on the first beat of a bar, are learnt as two temporally separate sequential events. Nevertheless, this encoding scheme allows for some temporal structure to be learnt, and for templates of pieces consisting of a fixed number of bars and phrases to be defined. Novel pieces are generated with a *random walk*[9] sampling method, with a *generate-and-test*[10] method ensuring generated events conform to the chosen temporal template. This initial exploration into statistical learning and generation of harmony concludes that the task is possible, although temporal constraints are necessary to prevent trivially short high probability pieces, and higher order structure such as functional cadential patterns are difficult to capture. Further n-gram studies of harmony model root progression theories in early tonal music (Hedges & Rohrmeier, 2011), and statistical structure in Bach chorales (Rohrmeier & Cross, 2008), both of which sample pitch class sets (pcsets) at the crochet (quarter note) level and take the most consonant pcset from each sampled segment.

[9] A random walk simply generates an event from a Markovian probability distribution, appends the event to the context before generating the next event with the new context. The process is repeated a fixed number of times, or until and an *end* symbol is generated.

[10] Generate-and-test methods are applied to random generation algorithms to ensure generated events conform to some template or constraints. Generation is carried out as normal without reference to the constraints, the generated event is then checked against the constraint and kept if it conforms, or discarded and the process repeated if it does not conform.

Another class of statistical models applied to tonal harmony are probabilistic graphical models.[11] Raphael and Stoddard (2004) present an HMM approach to functional harmonic analysis of polyphonic Musical Instrument Digital Interface (MIDI) data. The task is to label regular fixed-length segments of the input data with tonal centre, mode, and chord scale degree labels. Treating the pitch sets as the observed states, and the target labels as hidden states, standard HMM inference techniques are applied to find the optimum path of hidden states for the pitches observed in the MIDI data. Empirical evaluation is not carried out owing to the difficulty in defining a ground truth for harmonic analysis, however, the hand selected analyses are reported to be promising. Given that an HMM graph assumes statistical independence between observed states (pitch sets), it is noted that such a model is unable to take into account voice leading, and so more advanced graphical models are proposed. Extending the HMM approach to Bach chorale harmonisation, Allan and Williams (2005) use the Viterbi algorithm to find the best sequence of hidden states, representing the pitches of three lower voices relative to the melody and a harmonic function symbol, given the sequence of observable states, representing the melody pitches. Ornamentation (passing notes, suspensions, etc.) are added with a further HMM model. An analysis of negative log probabilities over chord states show that an HMM better accounts for statistical structure than simpler models only taking into account transitional probabilities of chord symbols, or individual chord symbols given only the corresponding melody notes, or chord symbols assuming statistical independence. The model produces reasonable simple harmonisations, although struggles with voice leading; in particular, the bass line is found to be unidiomatically disjointed. The predictive performance of other probabilistic graphical models including HMMs, Dynamic Bayesian Networks (DBNs), and feature-based DBNs is assessed by Rohrmeier and Graepel (2012), finding a DBN incorporating mode to perform best.

Paiement (2008) argues that the above approaches incorrectly simplify harmonic structure by only taking into account first order transitions between chord symbols. A more advanced probabilistic graphical modelling approach is proposed for jazz chord sequences, specifically aiming to model higher order structure. Chords are represented by their timbral content as a continuous vector of length 12 based on the relative strengths of each pitch class (explicitly present or in the harmonic overtone series) in the chord. From this representation a euclidean distance metric can be defined to model chord similarity. Global dependencies are captured by a graphical model with a three level structure (Figure 2.1). The highest level consists of several hidden layers representing metrical structure and accounting for global dependencies. The second level is a single layer for local dependencies, and the third level the continuous surface observations as modelled by a mixture of Gaussians. A final fourth layer can be added to account for

[11] Technically, a Markov model can also be construed as a trivial probabilistic graphical model.

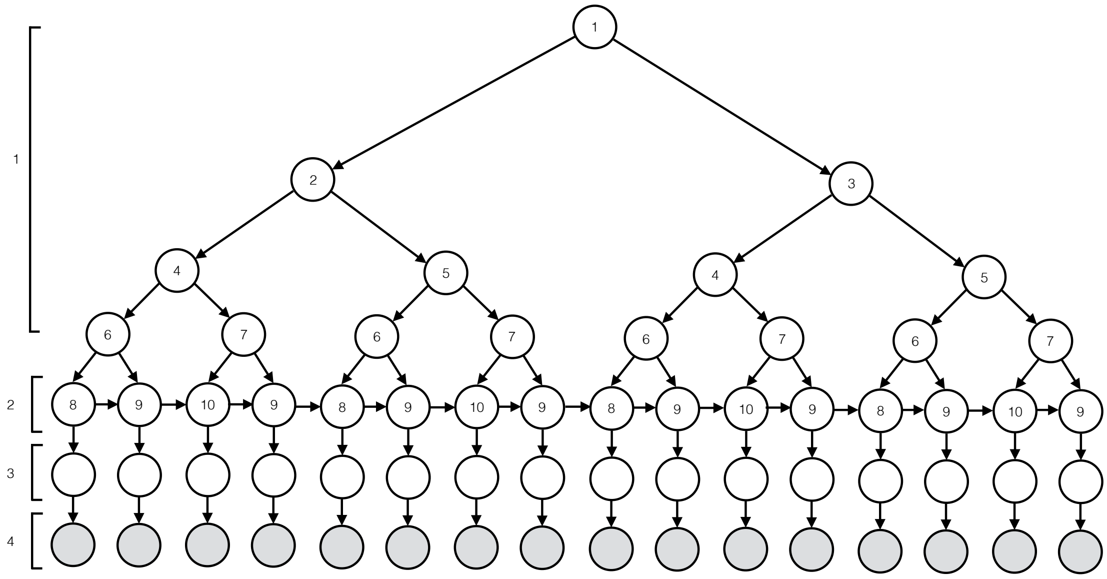


Figure 2.1: A probabilistic graphical model for chord sequences derived from Paiement, Eck and Bengio (2005, Figure 2). White nodes represent hidden states, whilst observable states are shaded grey. The bottom 16 nodes of level 4 represent chords each half a bar long.

chord substitutions (Paiement, Eck & Bengio, 2005). Overall, the probabilistic graphical model forms a Bayesian Network, with the level 1 hidden nodes forming a binary tree derived from the metrical structure of a typical 8-bar phrase. Numbered hidden nodes indicate nodes which share conditional probability parameters. However, linking the higher order structure of chord sequences to a typical binary metrical system prohibits probabilistic influence across bar lines and 4-bar phrases at certain levels in the hierarchy. This simplification of structure is potentially problematic bearing in mind that harmonic segments and units in jazz music frequently span such boundaries (c.f. Granroth-Wilding, 2013; Pachet, 2000; Steedman, 1984).

2.4 Multiple Viewpoint Systems

Multiple viewpoint systems (Conklin & Witten, 1995) are a representational framework and statistical modelling scheme for symbolic sequences. Multiple viewpoint systems are able to represent and capture statistical structure in multidimensional phenomena, such as music. Different dimensions, or features, of music are represented by *viewpoints*. Each viewpoint is associated with its own partial function, selecting a dimension or abstraction from a musical surface, and a finite-context model. Statistical structure is found through multi-dimensional, variable order Markov models, combining probability distributions from various models with information theoretic weighting schemes. Long and short term structure is captured with separate models and their respective probability distributions combined with the same method that combines viewpoint predictions. A full, technical, and formal description of multiple viewpoint systems is reserved for Chapter 3. In the meantime, the following section outlines the wide application of multiple viewpoint systems in various research areas, including music cognition and perception, computational musicology, and computational creativity.

2.4.1 Melodic Modelling

Early research developed multiple viewpoint systems tasked with modelling both monodic and two-voice Gregorian chant voice melodies (Conklin & Cleary, 1988). A high level overview of multiple viewpoint systems identifies viewpoints modelling absolute pitch, relative pitch (as an interval between both adjacent and vertically aligned notes), pitch class, pitch range, and duration. Fixed order Markov models are applied to learning statistical structure from the training data. Generation is proposed as an evaluation method, judging the model's ability to generate unique, stylistically recognisable sequences with reference to the training data. Sequences generated by random walk with a fixed order bound of 2 produce novel, stylistically recognisable, albeit unstructured chants. Longer fixed order bounds generate exact repetitions from the training data, whilst short fixed order bounds and uni-gram models generate disjointed, 'random' melodies.

Multiple viewpoint systems are further developed and formalised, and applied to predicting and generating Bach chorale melodies (Conklin, 1990; Conklin & Witten, 1995). A method for combining probability distributions with a weighted arithmetic mean, weighting by *Shannon* entropy (Shannon, 1948) is introduced (Conklin & Witten, 1995). An information theoretic evaluation metric, average entropy,[12] identifies a hand-crafted multiple viewpoint system comprised of four viewpoints as the best description of the test data of 5 chorales given a training set of 95 chorales, returning 1.87 bits/symbol. Subsequent improvements in multiple viewpoint modelling shows that a weighted geometric mean model combination method outperforms a weighted arithmetic mean (Pearce, Conklin & Wiggins, 2005), and establishes a computational method for constructing multiple viewpoint systems (Pearce, 2005, ch. 7) that outperforms the hand-selected systems of Conklin and Witten (1995). Additionally, the finite context modelling techniques employed have been extended with various smoothing methods to tackle problems associated with fixed-order models including the zero-frequency problem (Witten & Bell, 1991). Empirical testing over large datasets of chorale and folk

[12] Equivalent to mean information content in the current research (see §3.4.2).

song melodies show an escape method commonly referred to as Witten-Bell smoothing (or escape method C: Moffat, 1990; Witten & Bell, 1991) to be generally successful at predicting monophonic melodies (Pearce & Wiggins, 2004).

2.4.2 Representing Vertical Structure and Polyphony

The multiple viewpoint framework is extended to account for the representation of multiple simultaneous voices, and hence, vertical pattern discovery, by Conklin (2002). The framework has been formalised by Bergeron and Conklin (2011), and extended for the tasks of statistical learning, symbolic prediction, and music generation by Whorley (2013). The following provides a summary of this framework.

Firstly, an ontology of the musical objects making up the framework defines *Note*, *Seq*, and *Sim* objects. Each of these objects must be comprised of (at least) *duration* and *onset time* attributes. Typically, objects of type *Note* will consist of pitch class and octave attributes, or equivalently, a MIDI number. *Seq* (standing for sequence) and *Sim* (simultaneity) are polymorphic and may be comprised of any number of *Note*, *Seq*, or *Sim* objects. Music objects, *M*, may be defined by the algebraic data type:

$$\begin{gathered} M ::= Note|Seq(M)|Sim(M) \\ join : Seq(X) \times X \to Seq(X) \\ layer : Sim(X) \times X \to Sim(X). \end{gathered}$$

The function *join* concatenates musical objects of type *X* which do not overlap (as determined by their *duration* and *onset time*) to produce a *Seq* of *X's*, denoted by the use of square brackets, [...]. The function *layer* forms *Sim* objects by grouping objects with identical *onset times* (not necessarily *durations*), which are denoted with angled brackets ⟨...⟩.

The most basic, score-based method of encoding polyphonic music is as several sequences of notes, with all sequences commencing simultaneously, formally: $Sim(Seq(Note))$. In reference to the chorale extract in Figure 2.2, this would produce the following structure (labelling *Note* objects by pitch class only):

$$\begin{gathered} \langle [G, C, B\flat, A\flat, G, F, G]\,, \\ [E\flat, E\flat, E\flat, E\flat, F, F, E\flat, E\flat, D, E\flat]\,, \\ [B\flat, A\flat, B\flat, C, B\flat, B\flat, C, B\flat, B\flat]\,, \\ [E\flat, A\flat, G, C, D, E\flat, A\flat, B\flat, E\flat]\rangle. \end{gathered}$$

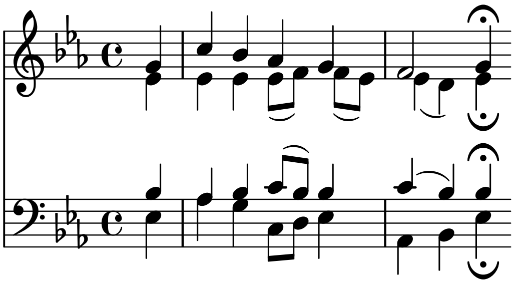


Figure 2.2: Opening phrase of four-part chorale: *"Ich will hier bei dir stehen"* by J.S.Bach.

Unfortunately, for the tasks of pattern finding and statistical learning, such an encoding is not a natural way of finding vertical patterns, since simultaneous *Note* objects can only be found after calculating durations and onset times along each sequence. Instead, Conklin (2002) proposes encoding polyphony as a $Seq(Sim(Note))$ structure; as a sequence of simultaneous note groups. This requires various *partitioning* methods to be considered. A *natural partitioning* starts a new $Sim(Note)$ group whenever all voices have the same onset:

$$\begin{gathered}[\langle [G], [E\flat], [B\flat], [E\flat] \rangle, \\ \langle [C], [E\flat], [A\flat], [A\flat] \rangle, \\ \langle [B\flat], [E\flat], [B\flat], [G] \rangle, \\ \langle [A\flat], [E\flat, F], [C, B\flat], [C, D] \rangle, \\ \langle [G], [F, E\flat], [B\flat], [E\flat] \rangle, \\ \langle [F], [E\flat, D], [C, B\flat], [A\flat, B\flat] \rangle, \\ \langle [G], [E\flat], [B\flat], [E\flat] \rangle].\end{gathered}$$

However, this partitioning method is prone to under-partitioning music; vertical structure may be missed if the harmonic rhythm is not aligned with grouping resulting from the common onset times. For example, beats 1 and 2 of the final bar of Figure 2.2 represent two distinct, consonant, harmonies, $ii^6_3 - V$, but are grouped in the same partition. The partitioning method developed for the multiple viewpoint framework is *full expansion.* Under a full expansion an onset in any voice starts a new $Sim(Note)$ group, splitting *Note* objects as necessary (Figure 2.3). The resulting $Seq(Sim(Note))$ structure is as follows:

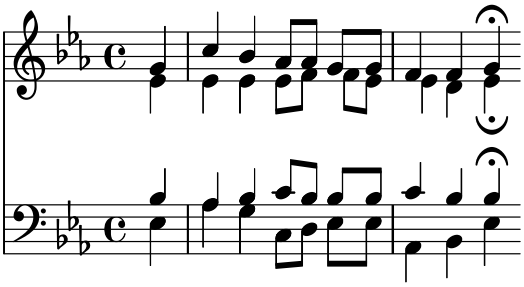


Figure 2.3: Full expansion of the opening phrase of four-part chorale: *"Ich will hier bei dir stehen"* by J.S.Bach.

$$
\begin{gathered}
[\langle[G],[E\flat],[B\flat],[E\flat]\rangle,\\
\langle[C],[E\flat],[A\flat],[A\flat]\rangle,\\
\langle[B\flat],[E\flat],[B\flat],[G]\rangle,\\
\langle[A\flat],[E\flat],[C],[C]\rangle,\\
\langle[A\flat],[F],[B\flat],[D]\rangle,\\
\langle[G],[F],[B\flat],[E\flat]\rangle,\\
\langle[G],[E\flat],[B\flat],[E\flat]\rangle,\\
\langle[F],[E\flat],[C],[A\flat]\rangle,\\
\langle[F],[D],[B\flat],[B\flat]\rangle,\\
\langle[G],[E\flat],[B\flat],[E\flat]\rangle].
\end{gathered}
$$

A full expansion is more likely to find all harmonic groups, at the cost of over partitioning, whereby some complete harmonic units are unnecessarily split (for example, bar 2, beat 4). To an extent, the undesirable effects of over partitioning can be reduced with *threaded viewpoints*,[13] a class of viewpoint that models non-adjacent events at specified temporal intervals, for example, at the bar or tactus levels. A boolean viewpoint, usually named `start` or `cont` can be used to distinguish between events that have true onsets (i.e. an onset both in the original and fully expanded versions) and events which are continuations of a previous event (Bergeron & Conklin, 2011; Whorley, 2013).

Various issues arise when extending multiple viewpoint systems tasked with predicting sequences from monophonic to polyphonic music. Firstly, the domain (or alphabet) size of viewpoints which span multiple voices is the Cartesian product of the alphabets of the individual viewpoints, creating alphabets which are impractically large in comparison with monophonic melodies. Whorley, Wiggins, Rhodes and Pearce (2013b) propose a method to address this issue by including only chords in the test corpus before aug-

[13] A detailed formal description of threaded viewpoints is reserved for §3.3.2.

menting the alphabet with chord transpositions within a defined range. The order in which voices are predicted has a significant impact on model performance, and therefore is best conducted in an informed manner as a series of sub-tasks (Whorley, Rhodes, Wiggins & Pearce, 2013a). Finally, the number of potential multiple viewpoint systems is far larger for polyphonic prediction in comparison to monophonic. This prompts practical considerations when automatically constructing locally optimal multiple viewpoint systems;[14] potential viewpoints added to a system must be pruned from the search at each iteration if they, for example, are found to perform poorly in a previous iteration (Whorley, 2013, p. 125).

2.4.3 Modelling Music Perception and Cognition

Multiple viewpoint modelling forms the core of the IDyOM model (Pearce, 2005; Pearce & Wiggins, 2012). IDyOM fulfils the role of a computational model for cognitive processes in a line of research chiefly studying melodic expectation and its relation with information theoretic properties of music. The methodological approach follows that of Desain et al. (1998) in implementing a cognitive theory as a computational model and providing an empirical, psychological validation. An algorithm models the mental process of interest, both algorithm and mental process exhibit a measurable behaviour for certain tasks, with agreement between the behaviours validating the model. A computational implementation holds distinct advantages over ‘pen-and-paper’ models (Rohrmeier, 2010; Wiggins, 2011), namely that the process of implementing and testing the model allows precise hypotheses to be made which may be empirical falsified (Popper, 1934). Furthermore, computational implementations may be adjusted and re-tested in reaction to experimental results, improving the model and understanding of the cognitive theory. IDyOM is such a computational model.

A series of studies investigate the extent to which a cognitive theory rooted in statistical learning, Markov modelling, and information theory can account for human behaviour when processing melodic expectations. Pearce and Wiggins (2006) seek to correlate the results of existing perceptual studies of melodic expectancy from three experiments with increasingly complex melodic contexts (Cuddy & Lunney, 1995; Manzara et al., 1992; Schellenberg, 1996), with predictions made in the form of probability distributions derived from IDyOM. IDyOM accounts for 72% of the variance for goodness-of-fit ratings for probe tones following a single melodic interval (Cuddy & Lunney, 1995), 83% of the variance for goodness-of-fit ratings for probe tones following melodic folk song phrases (Schellenberg, 1996), and 63% of the variance for entropy estimates of each note

[14] This process is known *viewpoint selection*, see §3.5

in chorale melodies (Manzara et al., 1992). In comparison with a two-factor implementation of Narmour's (1990) *Implication-Realization* model (Schellenberg, 1997), IDyOM subsumes the key bottom-up functions of *proximity* and *reversal* (Schellenberg, 1997) for all three experiments, rendering such innate, rule-based functions redundant in an account of human melodic expectancy. The high level of agreement between IDyOM and human behaviour in melodic expectation is upheld when tested with a new paradigm that aims to minimize interruption in the melodic stimuli with a 'clock face' visual cue for participants signalling when they are required to give expectation ratings for tones (Pearce et al., 2010c). Further electrophysiological results show low probability notes elicit a larger event-related potential (ERP) at a 400-450ms time period, a 14-30Hz beta-band oscillation over the parental lobe, and long range phase synchronisation between various regions.

An information theoretic understanding of melodic expectation is further enhanced by Bailes, Dean and Pearce (2013), who study the effects of recently integrated destabilising tonal information, and time delays on probe tones. Stimuli of contrasting tonal stability consisting of five chords and three notes are presented, followed by a probe tone with a delay of 0 to 19.2 seconds. Tonal stability is judged by the degree to which the three-note melodic continuation violates the tonal context established by the five chords. This can be measured empirically as the mean information content of the final three notes as calculated by IDyOM using the established tonal centre of the first five chords for an viewpoints modelling scale degree. A significant correlation was found between probe delay and probe pitch for both more and less unstable stimuli. Additionally, for the more unstable stimuli a correlation was found between pitch probabilities (calculated by IDyOM) and probe ratings for 6 second delays (but not for shorter ones). Finally, the rating profile of the probe tone after a maximal delay of 19.2 seconds is shown to be better matched by IDyOM's probability distribution compared to a Krumhansl-Kessler (Krumhansl & Kessler, 1982) frequency-based probability distribution (see §2.2.2). Overall, it is suggested that retrospective analysis plays a role in delayed expectations converging towards patterns which consolidate probabilistic information. Melodic expectation can also be understood from the perspective of predictive uncertainty, rather than perceived expectedness. In this respect, IDyOM models levels of perceived uncertainty with some degree of success, finding a weak correlation with the Shannon entropy (Shannon, 1948) of a probability distribution (Hansen & Pearce, 2014).

Statistical learning and melodic prediction play a role in furthering understanding in the underlying causes of congenital amusia[15] (Ayotte, Peretz & Hyde, 2002), a developmental disorder primarily manifesting itself in pitch perception and production

[15] Commonly known as tone deafness.

difficulties. Omigie, Pearce and Stewart (2012) and Omigie, Pearce, Williamson and Stewart (2013) present two experiments to assess the extent to which congenital amusia can be accounted for by difficulties in either forming, or consciously accessing statistical regularities in tonal music. Two experiments (Omigie et al., 2012) assess implicit (measuring response time in a forced choice timbre-discrimination of a cued target note) and explicit (rating degree of expectedness of a target note) components of melodic expectation. The probability of target notes is empirically controlled by IDyOM using viewpoints representing scale degree and melodic interval information. In the implicit task a melodic priming effect is found in both amusia and control groups, whereby high probability notes are processed quicker, eliciting a faster response time. However, in general, the amusia participants were both slower and less accurate than the control group. In the explicit task amusia participants struggled to identify high from low probability notes through explicit ratings in comparison with the control group. Nevertheless, surprisingly amusia participants were, to a small extent able to explicitly identify high from low probability notes possibly suggesting congenital amusia is not simply a categorical phenomenon with regard to conscious access to musical structure. A related ERP study (Omigie et al., 2013) shows a reduced early frontal negativity in amusia participants in contrast to controls when processing expected (low probability) pitches. It is suggested that such a response correlates with explicit knowledge of musical expectedness.

IDyOM is successfully applied as a computational model of melodic expectation in relation to various other music-related cognitive tasks, namely segmentation (Pearce et al., 2010b), memory (Agres et al., 2017), listening in live performance (Egermann et al., 2013), and simultaneous melodic and linguistic processing (Carrus, Pearce & Bhattacharya, 2013). Pearce et al. (2010b) tests IDyOM as a cognitive model of melodic segmentation, selecting segment boundaries at low probability events. As an unsupervised, statistical learning model, IDyOM performs comparably to innate (human-coded), rule-based models (Cambouropoulos, 2001; Frankland & Cohen, 2004; Lerdahl & Jackendoff, 1983; Temperley, 2001) in accounting for clustered melodic segmentations from 25 participants. Agres et al. (2017) show information theoretic properties (information content, coding gain, predictive information) are significant predictors of auditory sequence memorability. Stimuli are generated from a first order Markov model, systematically controlling the various information theoretic measures. IDyOM is used as a computational simulation, capturing within and across stimuli implicit learning, accurately accounting for 74% of the variance in expectedness ratings, and 85% in memory performance during the final experiment session.

Egermann et al. (2013) presents a study investigating the role of musical expectation in a live flute performance. Melodic pitch expectations predicted by IDyOM were found to correlate with both explicit ratings and emotional responses as measured by explicit valence and arousal ratings, facial electromyography (EMG), and peripheral arousal. Carrus et al. (2013) finds an interaction between syntactic and/or semantic language violations, and low or high probability notes (calculated by IDyOM) in an EEG study. Notably, a reported left anterior negativity present for syntactic violations is reduced when presented with low, in contrast to high, probability notes.

In summary, IDyOM, a statistical learning model built on multiple viewpoint representation schemes, has been shown to be a strong model for melodic expectation and, to a lesser extent, uncertainty. As musical expectation is a core component of musical cognition, IDyOM has also proven to be a useful model for other cognitive tasks such as segmentation, and memory recall.

2.4.4 Computational Musicology and Music Information Retrieval

The representational and probabilistic elements of multiple viewpoint systems have been applied to various computational musicology and music information retrieval tasks. The research reviewed in this section focuses on pattern discovery, classification, and segmentation.

The general principles of pattern discovery with multiple viewpoint systems are proposed in Conklin and Anagnostopoulou (2001). A potential pattern, P, is a sequence of elements in a given type, for example, pitch, pitch interval, or pitch contour. A pattern discovery algorithm aims to find patterns that occur significantly more frequently than their expected frequency. First, the probability of a pattern occurring in a corpus is calculated with a blended first- and zero-order Markov model (Conklin & Witten, 1995), and denoted by $p(P)$. The expected frequency (Equation 2.1) of a pattern, $E(P)$, is found by multiplying this probability by the number of possible locations where the pattern might potentially occur in the dataset.

$$E(P) = p(P) \times \Big(\#(\varnothing) - n \times (l(P) - 1)\Big) \tag{2.1}$$

Above, $l(P)$ is the length of a pattern, $\#(P)$ is a function returning the total count of a pattern in a corpus consisting of n pieces, $\varnothing$ is the empty pattern (i.e. $l(\varnothing) = 0$), and $\#(\varnothing)$ is equivalent to the total number of events in the corpus ignoring those that are undefined for the given type (e.g. pitch interval for the first event of a piece).[16] A

[16] An implicit assumption is made that the length of the pattern is not greater than the length of any pieces in the corpus. If this is the case, then $n \times (l(P) - 1)$ no longer represents the number of positions

pattern score, $S(P)$, is then defined by Equation 2.2.

$$S(P) = \frac{\Big(\#(P) - E(P)\Big)^2}{E(P)} \tag{2.2}$$

The statistical significance of a pattern is reported by a p-value representing the probability that a greater or equal pattern score may occur in a random viewpoint sequence using an exponential probability distribution. Longest significant patterns are identified by finding all significant patterns ($p < 0.01$), which occur in at least k (typically 10) pieces in a corpus, and removing all patterns which are subsumed by other significant patterns found in the corpus. A number of musically interesting patterns in various voices of a Bach chorale corpus are found, adjusting k in order to return a useful number of longest significant patterns. Conklin (2002) extends this approach to finding vertical (harmonic) patterns in the Bach chorale corpus (see §2.4.2 for an account of representing polyphony with the multiple viewpoint framework). An analysis returns 32 shortest significant patterns, many of which outline musicologically convincing cadential figures.

Conklin and Anagnostopoulou (2001) and Conklin (2002) make use of a Markov model as a probabilistic *background model*, against which significant patterns are identified. An alternative solution to a background model is to explicitly employ a corpus (signified by $\oplus$) and *anticorpus* ($\ominus$) in pattern discovery algorithms (Conklin, 2010; Conklin & Anagnostopoulou, 2011). The distinctiveness of a pattern, $\triangle(P)$, is defined as the ratio between the probabilities of the pattern occurring in the corpus and anticorpus (Equation 2.3). This may also be expressed as the ratio between the number of pieces the pattern occurs in in the corpus, and the expected number of pieces the pattern occurs in calculated with the estimated probability of the pattern occurring in the anticorpus. $c^{\oplus}(P)$ and $c^{\ominus}(P)$ are the number of pieces that contain the pattern P in the corpus and anticorpus respectively, and $n^{\oplus}/n^{\ominus}$ the total number of pieces in the corpus/anticorpus.

$$\begin{aligned} \triangle(P) &\overset{\text{def}}{=} \frac{p(P|\oplus)}{p(P|\ominus)} \equiv \frac{c^{\oplus}(P)}{p(P|\ominus) \times n^{\oplus}} \\ p(P|\oplus) &= \frac{c^{\oplus}(P)}{n^{\oplus}} \\ p(P|\ominus) &= \frac{c^{\ominus}(P)}{n^{\ominus}} \end{aligned} \tag{2.3}$$

that the pattern cannot possibly occur in the corpus, and therefore, $\#(\varnothing) - n \times (l(P) - 1)$ is no longer the number of positions that the pattern might possibly occur in the corpus.

Patterns that exceed a specified threshold in $\triangle(P)$ are said to be *distinctive*. A *maximally general distinctive pattern* is a pattern that is subsumed by no other distinctive patterns. Conklin (2010) identifies distinctive patterns in folk song melodies, and chord sequences. Both domains consist of three genres: when selecting one genre as the corpus, the other two are labelled as the anticorpus. When searching patterns any viewpoint can be used to match any event in a sequence, meaning that different levels of representational abstraction are in play across a pattern subsequence. The maximally general distinctive patterns identified are musically interesting, however, in some cases can be trivially general: for example, simply a rising pitch interval of six semitones for Austrian folk songs. Different partitionings of folk song corpora are considered by Conklin and Anagnostopoulou (2011), namely geographic regions and song function or type. Additionally, the representational viewpoint is restricted to melodic pitch interval, from which some definitive maximally general distinctive patterns are identified. Further work in this area of research includes formalizing pattern subsumption for vertical viewpoints (Bergeron & Conklin, 2011), and discovering antipatterns (patterns which are notably absent from a given corpus) in Basque folk tunes (Conklin, 2013a).

The probabilistic components of multiple viewpoints systems have been applied to classification tasks in melodic and harmonic domains (Conklin, 2013b; Hedges, Roy & Pachet, 2014). Given training sets of length T associated with a set of classes, C, and probabilistic models capable of calculating $p(e_1^T|c)$; the probability of a sequence e_1^T indexed 1 to T, given a class-specific trained model c, it is possible to select the most probable class, c^*, for a given sequence (Equation 2.4).

$$\begin{aligned}
c^* &= \underset{c \in C}{\arg\max} p\left(c|e_1^T\right) \\
p\left(c|e_1^T\right) &= \frac{p\left(e_1^T|c\right) \cdot p(c)}{p\left(e_1^T\right)} \\
p\left(e_1^T\right) &= \sum_{c \in C} p\left(e_1^T|c\right) \cdot p(c)
\end{aligned} \tag{2.4}$$

Classification with different multiple viewpoint representations is carried out by calculating $p\left(e_1^T|c\right)$ for each viewpoint, and combining the probabilities with an unweighted geometric mean. Using this technique Conklin (2013b) classifies folk songs by genre with accuracies of 77.6%/88.7% for Basque/European corpora, and classifies with accuracies of 58.8%/79.2% for the two corpora over geographical region. These results compare favourably with Support Vector Machine classifiers using global features (Hillewaere, Manderick & Conklin, 2009), as well as edit distances, compression distance, and string subsequence kernel methods (Hillewaere, Manderick & Conklin, 2012) in identical classification tasks. A similar multiple viewpoint classification method is applied in Hedges et al. (2014), classifying lead sheets by composer, sub-genre, performance style, and meter. The multiple viewpoint representation allows for both harmonic and melodic representations to be taken into account. The best performing classifier for a 9-class composer classification combines melodic, harmonic, and temporal information, returning an accuracy of 67.3%, outperforming a simpler Markovian classifier (Perez-Sancho, Rizo & Inesta, 2009). The classifier is developed to find the maximal length classified subsequences of chords in a jazz lead sheet, producing a musicological analysis which dynamically tracks composer styles throughout a composition.

The probabilistic components of multiple viewpoint systems are the underlying mechanisms in applying IDyOM to the music information retrieval task of segmenting melodic sequences (Pearce, Mullensiefen & Wiggins, 2010a). A peak picking algorithm is defined to select segment boundaries that coincide with rises in *information content* (MacKay, 2003), in other words, notes with a low probability relative to their context. The IDyOM segmentation model is tasked with segmenting at the phrase level 1,705 annotated Germanic folk song melodies, and its performance compared to various rule-based models (Cambouropoulos, 2001; Frankland & Cohen, 2004; Lerdahl & Jackendoff, 1983; Temperley, 2001). An F1 score of 0.58 places it amongst the top four of nine segmentation models tested, outperformed only by hand-coded rule based systems, with a highest F1 score of 0.66 achieved by a hybrid model. From a musicological and psychological perspective on a smaller, but more detailed, scale, Wiggins (2010) presents a cue abstraction (Deliege, 1987) and paradigmatic analysis (Ruwet, 1972) of Debussy's *"Syrinx"* for flute. An IDyOM model predicting pitch and durations separately, using only the previous notes in the piece itself as training data, is able to identify structurally salient segments by finding peaks in information content profiles. The paper aims to work towards an explanatory model of human listening and information processing when experiencing the piece.

For pattern discovery, classification, and segmentation the multiple viewpoint representation allows different levels of abstraction to be considered simultaneously, and dynamically. This is a powerful tool in symbolic music information retrieval tasks where the level of abstraction can be a restrictive issue, as it is often not apparent *a priori* which levels abstractions are optimal, or indeed appropriate.

2.4.5 Generation and Computational Creativity

Conklin (2003) presents a general discussion on the topic of applying statistical models to music generation. Music generation is presented as a sampling task from a statistical model, which embodies the stylistic structure of a target corpus. Four approaches to sampling are outlined: random walk selects the most probable continuation from a Markov model at each event, applying Viterbi decoding (Viterbi, 1967) to find the most probable sequence from an HMM or first-order Markov model, stochastic sampling techniques such as *Gibb's* and *Metropolis* sampling, and pattern-based sampling applying stochastic sampling methods that conserve pattern structure.

Pearce and Wiggins (2007) use the *Metroplis-Hastings* algorithm (MacKay, 1998) to generate chorale melodies, with IDyOM as the underlying probabilistic model. The algorithm runs for a pre-defined large number of iterations from an initial state, which may be a random sequence or an existing chorale. An event is selected at random, and its pitch changed by sampling from a probability distribution calculated by IDyOM representing the continuation of the event given the preceding context. The change is not accepted automatically, but instead accepted with probability:

$$min\left[1, \frac{p_m(s'_k) \cdot p_m(t^i|t_1^{i-1})}{p_m(s_k) \cdot p_m(t'^i|t_1^{i-1})}\right].$$

The posterior probability of the sequence for iteration k given the multiple viewpoint model m, is $p_m(s_k)$, and the probability of an event t at index i given its context t_1^{i-1} is $p_m(t^i|t_1^{i-1})$. The proposed sequence and event are signified as s' and t' respectively. Quantitative and qualitative feedback from 16 judges in a controlled listening experiment revealed the generated chorales not to be especially stylistically typical of the corpus, specifically in terms of tonal and phrase structure, and to a lesser extent, melodic structure.

Multiple viewpoint techniques for harmonising four-part chorales with random walk methods are presented in Whorley (2013, ch. 9). Polyphonic generation is a substantially more challenging task than monophonic generation; not only are four stylistically recognisable voices required to be generated, but also their contrapuntal interactions stylistically and syntactically (Mann, 1965) acceptable, as well as creating a coherent harmonic structure. These complex interactions are captured with a viewpoint system which permits linking both within and between voices, and the generation task simplified considerably by dividing into sub-tasks which first generate the bass voice given the soprano, before generating the remaining middle voices. In order to limit the knock-on impact of generating low probability events (which remain in the context of the subsequent notes generated) a probability threshold is employed below which events are not selected for generation. Generated harmonisations are assessed musicologically, and are overall found to be stylistically recognisable, albeit with some notable voice leading violations and harmonically uncharacteristic dissonances. It is worth noting that the system employs an entirely unsupervised statistical learning technique, without expert knowledge (c.f. Ebcı̈oğlu, 1986, 1990). Subsequent research (Whorley & Conklin, 2016) empirically assesses generation in terms of harmonic rule violations, finding them to correlate with high cross entropy (low probability) generations.

Random walk techniques often suffer from a lack of overall structure or direction in their generations. A number of approaches using multiple viewpoints have been employed to address this challenge. Conklin (2016) finds semiotic patterns to capture the cyclical structure of trance music, generating with a modified random walk technique which filters each step of the generative process to comply with the semiotic structure. Herremans, Weisser, Sörensen and Conklin (2015) generates music for *bagana* employing a variable neighbourhood search; a combinatorial optimization algorithm with hard constraints for structural coherence. A viewpoint measuring information contour (which can be considered as the inverse of a probability profile across a sequence) is defined to guide the objective function during search. Finally, Pachet and Roy (2011) present efficient constraint-based techniques for generating stylistically and structurally coherent musical sequences. Markovian sequences from a probabilistic model built from a training corpus can be given non-local structure by complying with user-defined unary constraints. Viewpoints are used as a flexible representational scheme, usually employed as single viewpoints. This approach has been developed to enforce metrical structure (Roy & Pachet, 2013), re-harmonize melodies (Pachet & Roy, 2014b), control the amount of exact repetition from a training corpus (Papadopoulos, Roy & Pachet, 2014), and generate both melodic and harmonic elements of a lead sheet (Pachet & Roy, 2014a).

2.4.6 Wider Applications

Multiple viewpoint systems are used in research relating to a wide variety of topics, some of which do not fall directly into those outlined in the preceding sections. An important quality of multiple viewpoint modelling is that the statistical learning techniques are general, and not specific to Western tonal music, or indeed to the musical domain. Multiple viewpoint systems have been applied to modelling Turkish folk music (Sertan & Chordia, 2011), as well as traditional North Indian music (Chordia, Sastry & Albin, 2010; Srinivasamurthy & Chordia, 2012). Symbolic linguistic sequences have been modelled using basic viewpoints representing phoneme and stress, for segmentation tasks at the syllable level (Wiggins, 2012a), in addition to morpheme, word, and phrase levels (Griffiths, Purver & Wiggins, 2015). Variations on the underlying statistical model have been explored, which traditionally are variable-order Markov models implemented as *trie* (Conklin & Witten, 1995) or *suffix tree* (Pearce, 2005) data structures. Triviño-Rodriguez and Morales-Bueno (2001) instead use a *prediction suffix tree* (Ron, Singer & Tishby, 1996), extended for multiple attribute prediction. Chorale melodies are modelled, and melodies generated with a sequential random sampling finding the most *representative* continuation node for a context at each step. Finally, Cherla, Weyde, Garcez and Pearce (2013) find that a Restricted Boltzmann Machine (RBM) is able to take advantage of longer contexts more efficiently than *n*-gram models in predicting a Bach chorale melody dataset.

To conclude, this section has summarised literature applying multiple viewpoint systems to various tasks. The powerful representational framework of multiple viewpoint systems enable them to model monophonic and polyphonic music more effectively than less sophisticated, single viewpoint representations. The information theoretic groundings of the theory mean that certain multiple viewpoint systems pose attractive *explanatory* models (Wiggins, 2007) of music perception and cognition. Other applications include music generation, pattern discovery, and segmentation in both music and language domains. The strength and breadth of research using multiple viewpoint systems is a strong motivating factor to develop and form deeper understandings of these models in the current research (Part II).

2.5 Positioning of the Current Research

The three main areas of research and literature reviewed in this chapter (music perception and cognition, computational models of tonal harmony, and multiple viewpoint systems) provide context for the current research. Expectation, particularly in the auditory domain, is established as a fundamental cognitive process, underpinned by evolution, capable of inducing valenced emotional states over a range of arousal levels. Statistical learning is proposed as an account of how accurate expectations and predictions are built up with repeated exposure to stimuli, with a substantial body of reviewed empirical research showing it to be a plausible account of various musical phenomena across melodic and harmonic domains. The current research, therefore, is situated among the learned (as opposed to non-learned, see §2.3) computational models, focusing on developing statistical and probabilistic methods to account for music cognition. In this sense, music's existence can be accounted for by cognitive processes (Wiggins et al., 2010), rather than quasi-platonic entities (Mazzola, 2002). However, a common valid criticism of learned models of harmony, in particular Markovian approaches, are that they do not sufficiently account for higher order structure (Rohrmeier, 2011). Whilst the work of Paiement (2008) does account for both non-local dependencies and higher order structure, the Bayesian Network has a fixed, inflexible structure. The models developed in Part III of the current research aim at building computational models capable of capturing arbitrary hierarchical structure, in a statistically driven, bottom-up manner.

As the primary musical domain of the current research, the reviewed cognitive processes underpinning an understanding of tonal harmony (§2.2.3) indicate that both expert and non-expert listeners posses the cognitive ability to perceive harmonic relatedness, transitional expectations, and (to a degree) hierarchical structure. Importantly an upper limit of around 10 seconds is found for cognitive effects of long-term dependencies in tonal harmony (Farbood, 2010; Woolhouse et al., 2016), strongly implying that recursive elements of tonal harmonic structure are bounded. Similar studies suggest humans are limited to comprehending only three or four levels of centre-embedded recursion in natural language speech and writing (Karlsson, 2007). Noting that bounded recursion can be satisfactorily approximated with finite-state grammars, the argument can be made that the ability to parse context-free grammars is not a requisite of understanding tonal harmony. The computational techniques developed in Part III of this thesis, therefore, aim at capturing higher-order structure with models based on finite-state grammars. Furthermore, the techniques developed are explanatory, rather than merely descriptive (Wiggins, 2007), using information theory as a driving force to model music cognition. As such, they aim to account for the underlying reasons for segmentation, higher-order structure, and expectation, in tonal harmonic sequences.

Finally, a comprehensive review of developments and applications of multiple viewpoint systems has been conducted, from which it is possible to identify a number of relatively unexplored avenues of research. Currently, there is only a limited understanding of how multiple viewpoint systems predict multiple target attributes; motivating the in depth analysis presented in Chapter 5. Additionally, the viewpoint selection algorithm presented in Pearce (2005, pp. 127-128) finds locally optimal viewpoint systems in an excessively large search space. Chapter 7 investigates the extent to which these search solutions can be considered optimal within a bounded search space. One of the main strengths of multiple viewpoint systems and statistical learning in general are that they are extremely general, and can be applied to multiple domains. The current research takes the opportunity to investigate in detail the application of multiple viewpoint systems to the domain of chord sequences, which have been relatively unexplored with multiple viewpoint techniques, Conklin (2010) and Hedges et al. (2014) excepted. The depth and variety of applications presented in §2.4 justifies Part II of the current thesis in enhancing and advancing a detailed understanding of multiple viewpoint systems as a statistical learning technique for symbolic, multidimensional, temporal sequences.

Chapter 3

Theoretical Background: Multiple Viewpoint Systems

3.1 Overview

Multiple viewpoint systems are a central modelling technique used throughout the thesis, with this chapter providing the background detail of the theory. The original motivations behind developing multiple viewpoint systems are outlined briefly in §3.2. §3.3 is a theoretical exposition of multiple viewpoint systems compiling the formal descriptions from the original thesis by Conklin (1990), the seminal paper of Conklin and Witten (1995), and subsequent works of Pearce (2005) and Whorley (2013). Subtleties between the works and formal description used for the current research are clarified. The key probabilistic calculations and performance metrics are presented in §3.4, an algorithm for selecting viewpoints in a system is summarised in §3.5, and the computational complexity of multiple viewpoint systems discussed in §3.6.

3.2 Motivations

The original motivations behind developing multiple viewpoint systems were to develop both predictive and generative statistical models of symbolic music (Conklin & Witten, 1995). Music is fundamentally a multidimensional entity, whether considered at the level of the sound wave, or at the symbolic level. A complex sound wave is highly multidimensional, requiring frequency component analysis before useful information can be extracted. At the symbolic level, most Western tonal music (the focus of this paper) can be described as a sequence of notes, which themselves consist of dimensions such as pitch, onset, duration, dynamic, and timbre. Multiple viewpoint systems aim to address some of the problems arising from modelling multidimensional symbolic sequences, in particular sparsity issues, whilst also taking advantage of useful traits such as correlation between dimensions. Additionally, multiple viewpoint systems present a framework which takes advantage of the fact that the basic event language can be fruitfully modelled in a language other than itself; for example, the pitches can not only be modelled in terms of pitches, but also pitch intervals. Multiple viewpoint systems combine the performance of individual expert models akin to *product of experts models* (Hinton, 1999, 2000), by weighting individual models by *Shannon entropy* (Shannon, 1948) giving more certain models increased influence. As a result, they are able to outperform a single model with the same information.

3.3 Formal Description

At their core, multiple viewpoint systems aim to estimate the probability of a sequence of events e_1^i indexed from 1 to i. Each event, e, has an internal structure consisting of a number of *basic attributes*, each drawn from a finite set of symbols making up their domain. Attributes (basic or otherwise) describe abstract properties of events, and are specified by their type, τ. The domain of τ represents the set of syntactically valid elements for that type, and is denoted by $[\tau]$. The set of all possible events is referred to as the event space, $e \in \xi$, and comprises the Cartesian product of the domains of all relevant basic types $\tau_{b_1}...\tau_{b_N}$:

$$\xi = [\tau_{b_1}] \times [\tau_{b_2}] \times ... \times [\tau_{b_N}]. \tag{3.1}$$

A *viewpoint*, as defined by Pearce (2005), is a partial function, $\Psi_\tau : \xi^* \rightharpoonup [\tau]$, which maps a sequence of events in ξ^* onto elements of type τ.[1] Originally, viewpoints have been defined to consist of both a partial function Ψ_τ, and a context model of sequences in $[\tau]^*$ (see Conklin, 1990; Conklin & Witten, 1995; Whorley, 2013). The present research necessarily[2] follows the approach of Pearce (2005) by viewing a viewpoint as purely a representational formalism, and thus separate the statistical model from the viewpoint definition.

[1] Note ξ^* denotes the Kleene closure of ξ, comprising all possible sequences composed of e including the empty sequence.

[2] See Chapters 10 and 11 where viewpoints are used as identity functions for chunks.

Following the music representation principles established by Lewin (1987) in the form of Generalised Interval Systems (GISs) and by Harris, Smaill and Wiggins (1991)[3] in the Common Hierarchical Abstract Representation for Music (CHARM) framework, τ is conceived of as an abstract data type complying with various addition and subtraction operations allowing, for example, durations, time points, pitches, and pitch intervals to be represented. Typically, $[\tau]$ often consist of integers or rational numbers (Conklin & Witten, 1995; Pearce, 2005; Whorley, 2013), whose mathematical properties map onto those of, for example, pitch and duration. This allows Ψ_τ to potentially be defined in terms of arithmetic operators, rather than specified functions operating over non-numerical symbols in $[\tau]$. The current research follows this approach, both to remain consistent with established definitions, and to allow Ψ_τ to be understood without defining functions requiring music-theoretic knowledge. For example, if [`cpitch`] is the set of all note names and octave combinations (e.g. $C\sharp 4$), $\Psi_{\texttt{cpint}}$ (Pearce, 2005) could be defined as the chromatic interval between sequential pitches, assuming enharmonic equivalence. However, more simply, if [`cpitch`] is the set of all integers in a range (representing a MIDI number), $\Psi_{\texttt{cpint}}$ is simply the current `cpitch` minus the previous `cpitch`. The semantic domain, $[\![\tau]\!]$, of type τ allows for separation between the implementation and representation schemes. A function $[\![\cdot]\!]_\tau : [\tau] \to [\![\tau]\!]$, maps from the domain of τ to the semantic domain of τ. A slight clarification absent in the works of Conklin (1990), Conklin and Witten (1995), Pearce (2005), and Whorley (2013) is useful at this point. Note that some elements of $[\tau]$ may map onto many elements in the human readable representation scheme, for example the MIDI note number 48 maps onto many note names including C_4, $B\sharp_4$, $D\flat\flat_4$, etc. Therefore, to avoid a one-to-many mapping, and in order for $[\![\cdot]\!]_\tau$ to be strictly functional, $[\![\tau]\!]$ must comprise of sets of semantically equivalent elements, and $[\![\cdot]\!]_\tau$ maps from individual elements in $[\tau]$ to a set of elements in $[\![\tau]\!]$.

Finally, each viewpoint has an associated *type set* (Conklin & Witten, 1995), $\langle\tau\rangle \subseteq \{\tau_{b_1}, ..., \tau_{b_N}\}$, denoting the set of basic types the viewpoint is derived from and, therefore, capable of predicting. For basic types this will simply be the basic type of itself. For convenience, Table 3-A summarises the sets and functions associated with viewpoints.

3.3.1 Creating Viewpoint Sequences

Sequences of events are converted into sequences of viewpoint elements with a *matching function*: $\Phi_\tau : \xi^* \rightharpoonup [\tau]^*$. This function converts sequences in ξ^* to sequences of viewpoint

[3] See also Wiggins, Miranda, Smaill and Harris (1993).

Table 3-A: Sets and functions associated with typed attributes.

Symbol	Interpretation	Example
τ	A typed attribute	`cpitch`
$[\tau]$	Syntactic domain of τ	$\{60, ..., 72\}$
$\langle\tau\rangle$	Type set of τ	$\{$`cpitch`$\}$
$\llbracket\tau\rrbracket$	Semantic domain of τ	$\{\{B\sharp_3, C_4...\}, ..., \{B\sharp_4, C_5\}\}$
$\llbracket\cdot\rrbracket_\tau : [\tau] \rightarrow \llbracket\tau\rrbracket$	Semantic interpretation of $[\tau]$	$\llbracket 60\rrbracket = \{B\sharp_3, C_4, ...\}$
$\llbracket\cdot\rrbracket'_\tau : \llbracket\tau\rrbracket \rightarrow [\tau]$	Syntactic interpretation of $\llbracket\tau\rrbracket$	$\llbracket B\sharp_3\rrbracket' = 60$
$\Psi_\tau : \xi^* \rightharpoonup [\tau]$	Viewpoint function	Projection function

Note. Replicated from Pearce (2005, Table 5.1) with slight modifications to the semantic domain and semantic interpretation.

elements, $[\tau]^*$, recursively skipping undefined symbols as shown in Equation 3.2[4].

$$\Phi_\tau\left(e_1^i\right) = \begin{cases} \varepsilon & \text{if } e_1^i = \varepsilon \\ \Phi_\tau(e_1^{i-1}) & \text{if } \Psi_\tau(e_1^i) = \perp \\ \Phi_\tau(e_1^{i-1}) \| \Psi_\tau(e^i) & \text{otherwise} \end{cases} \tag{3.2}$$

The inductive inference of the multiple viewpoint system adds sequences from e_1^i after conversion using Φ_τ to the relevant probabilistic model. In order to prevent the same sequence in $[\tau]^*$ being added more than once to the model, a check that $\Psi_\tau(e_1^i) \neq \perp$ must be made.[5] This is because if $\Psi_\tau(e_1^i)$ is undefined, $\Phi\tau(e_1^i)$ will equal $\Phi_\tau(e_1^{i-1})$ (Conklin & Witten, 1995).

3.3.2 Viewpoint Classes

Different classes of viewpoint are able to model patterns in sequences in a variety of ways. *Basic viewpoints* model the attributes that form the event; in previous research these are defined as those attributes immediately available from the representation of the data (Conklin & Witten, 1995). The Ψ_{τ_b} of a basic viewpoint is simply a projection function (Conklin, 1990, p. 60) selecting the relevant attribute from an event, and $\langle\tau\rangle$ contains only the basic type itself, τ_b.

Derived viewpoints are derived from one or more *basic viewpoints* by applying some operator, for example simplifying attributes through categorisation or modelling relationships between attributes of adjacent events. They allow rich relational qualities of sequences to be modelled, for example using pitch intervals to model pitch sequences. They can also be viewed as a way of abstracting information away from the basic event

[4] Note $\|$ indicates sequence concatenation.

[5] Note that the check to see if $\Psi_\tau(e_1^i)$ is undefined involves only the final (i^{th}) element for all viewpoints.

sequences in order to find more general patterns. The type set of a derived viewpoint is the set of types it is derived from.

Linked viewpoints model the interaction between *primitive viewpoints*,[6] combining viewpoints as direct products (Lewin, 1987) to form a conjunction of attributes. They are defined by a product type over n constituent types: $\tau = \tau_1 \otimes ... \otimes \tau_n$. The domain of a linked viewpoint is defined by: $[\tau] = [\tau_1] \times ... \times [\tau_n]$, and the typeset: $\langle \tau \rangle = \bigcup_{k=1}^{n} \langle \tau_k \rangle$. Finally, the viewpoint function of a linked viewpoint is as follows:

$$\Psi_\tau(e_1^i) = \begin{cases} \perp & \text{if } \Psi_{\tau_j}(e_1^i) = \perp \text{ for any } j \in \{1, ..., n\} \\ \big(\Psi\tau_1(e_1^i), ..., \Psi\tau_n(e_1^i)\big) & \text{otherwise.} \end{cases}$$

Test viewpoints are a subclass of derived viewpoints used to mark temporal locations in the event sequence with a boolean value. Although capable of predicting the basic type they are derived from, the primary function of test viewpoints are to assist in the construction of threaded viewpoints (discussed below).

Threaded viewpoints are able to model patterns between non-adjacent events, denoted by a *base viewpoint* and a test viewpoint: $\tau_{base} \ominus \tau_{test}$, with $\ominus$ indicating a threaded relationship between the viewpoints. The base viewpoint is a derived viewpoint (which may be linked) which would ordinarily be defined by adjacent events in the sequence (for example, viewpoints modelling pitch interval, rather than viewpoints modelling absolute pitch). The test viewpoint simply marks with a boolean specific locations in the sequence, for example the first beat of each bar, or the main tactus beats. A threaded viewpoint filters out events where the test viewpoint is false, allowing the relationships between non-adjacent events to be represented directly. By way of example, `thrbar` (Pearce, 2005), or `cpint`$\ominus$`FirstInBar`, models the chromatic pitch interval, `cpint`, between the first notes of successive bars. A further example can be found in Table 4-D (see §4.4.1.6) where `RootInt`$\ominus$`FiB` marks the chromatic interval class between the roots of chords on the first beat of each bar. Conklin and Witten (1995) defines $\Psi_\tau(e_1^i)$ for a threaded viewpoint to return a tuple containing the elements in the base viewpoint (e.g. [`cpint`]) and an inter-onset-interval, `ioi`, measuring the distance in timebase steps the current event and its predecessor (after the test viewpoint filtering).

[6] Any individual viewpoint that is not linked or threaded.

3.4 Probability Calculations and Performance Metrics

3.4.1 Finite Context Models

A multiple viewpoint system consists of a number of finite context models of sequences in $[\tau]^*$. m_τ denotes a finite context model for the viewpoint τ. Although in theory any finite context model can be used, almost[7] all multiple viewpoint systems to the author's knowledge use a form of Markov model, estimating the probability of a viewpoint element $t \in [\tau]$ using the *maximum likelihood*[8] as follows:

$$p\big(t^i \mid t_{i-n+1}^{i-1}\big) = \frac{c\big(t^i | t_{i-n+1}^{i-1}\big)}{\sum_{t \in [\tau]} c\big(t | t_{i-n+1}^{i-1}\big)}. \tag{3.3}$$

Here, the order of the Markov model is said to be $n-1$, alternatively, it can be described as an n-gram model. The frequency count of a symbol a given its context b is signified by $c(a|b)$.[9] *k-fold cross validation* is typically used to train and evaluate multiple viewpoint systems. For each fold, the dataset is divided into a held-out test set comprised of $1/k^{th}$ of the dataset, and training set comprised of the remainder of the dataset. The training set is used to build the model, counting n-gram sequences to estimate probabilities, whilst the test set remains unseen by the model. The probabilities of events in the test set given the trained model are then calculated to evaluate performance, usually with the mean *information content* (see §3.4.2).

Markov models using only a single fixed-order model are prone to poor predictions if the context (of length $n-1$) is too long to match sequences in the training data, or too short to make specific predictions. Conklin (1990) introduces using the Prediction by Partial Match (PPM) algorithm (Cleary & Witten, 1984) for estimating the conditional probability in multiple viewpoint systems. PPM utilises a technique referred to as *blending*, combining predictions from all orders up to an order bound to produce accurate predictions. Pearce and Wiggins (2004) test variations of PPM with various smoothing techniques (discussed in detail in §5.2.1), and note that the model can be implemented efficiently as a suffix tree using an online construction algorithm (Ukkonen, 1995).

[7] Cherla et al. (2013) is an interesting exception, replacing the usual Markov model with an RBM.

[8] Here, maximum likelihood refers to finding the parameters (probabilities assigned to conditional probability distributions) that maximises the probability of the training corpus only with no probability space reserved for new symbols. See Manning and Schütze (1999, pp. 197-199).

[9] Alternatively, the frequency counts could be written with sequence concatenation: $c(b||a)$.

3.4.2 Performance Metrics

Following a *natural language processing* approach (Manning & Schütze, 1999), the quality of a probabilistic model can be measured by the degree to which its probability function describes the data. *Information content* (MacKay, 2003) is a useful performance metric, representing an estimate of the number of bits required to describe an event drawn from a discrete probability distribution. Mean information content can also be a measure of the *cross entropy* between two distributions (Manning & Schütze, 1999, pp. 74-76), even when one probability distribution is unknown. In other words, mean information content represents the degree of fit between the probability distribution of the model and the true probability distribution of the stochastic process that generated the training data. The information content of a single event is given by Equation 3.4 and the mean information content of a sequence of length N by Equation 3.5. An information theoretically efficient model will return a low mean information content, resulting from relatively high probability estimates for events, suggesting a close fit between the model and statistical structure underlying the training data. Typically, the mean information content of a corpus is calculated with a k-fold cross validation (Conklin & Witten, 1995; Pearce & Wiggins, 2004), with the probability function $p\left(e^i|e_1^{i-1}\right)$ estimated from k training sets and the mean information content calculated from the corresponding testing sets.

$$h\big(e^i|e_{i-n+1}^{i-1}\big) = -\log_2 p\big(e^i|e_{i-n+1}^{i-1}\big) \tag{3.4}$$

$$\bar{h}\big(e_1^N\big) = -\frac{1}{N}\sum_{i=1}^{N}\log_2 p\big(e^i|e_{i-n+1}^{i-1}\big) \tag{3.5}$$

Mean information content is a useful performance metric for models, independent of their specific applications. The metric can be viewed as such for much of Part II of the current thesis, which deals with the predictive performance of multiple viewpoint systems in general. However, mean information content takes on a different role in Part III of the thesis, which presents an implementation of a cognitive architecture. Ultimately, the performance of the architecture as a model of cognition must be in its ability to simulate testable aspects of human behaviour (for example, expectation, segmentation, and lexical ambiguity, see Wiggins & Forth, 2015). However, comprehensive model evaluation cannot be carried out on an, as yet, undeveloped computational implementation of the cognitive architecture. Therefore, the present research (Part III) minimises mean information content as a heuristic for the purposes of model selection in developing the cognitive architecture, noting that, when tested, models with low information content correspond with models correlating closest with human data of musical expectation (Pearce & Wiggins, 2006).

3.4.3 LTM and STM

Conklin and Witten (1995) introduce the notion of a long-term model (LTM) and short-term model (STM) to capture global and local statistical structure respectively. The LTM is built from the held-out training set of the k-fold cross validation, whilst the STM is built dynamically on an event-by-event basis for each composition in the test set and then discarded after the composition has been processed. A third type of model, LTM+, merges qualities from both models, building both from the training set dynamically on an event-by-event basis. The current research follows both Conklin and Witten (1995) and Pearce et al. (2005) in combining predictions in two stages: first viewpoint predictions within the LTM(+) and STM separately, and second combining the predictions from the LTM(+) and STM themselves (Figure 3.1).

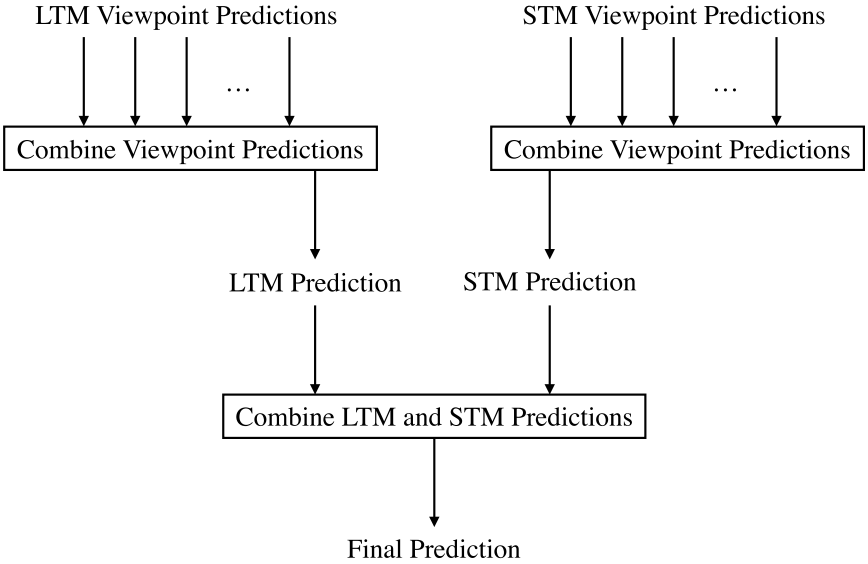

LTM Viewpoint Predictions
STM Viewpoint Predictions
...
...
Combine Viewpoint Predictions
Combine Viewpoint Predictions
LTM Prediction
STM Prediction
Combine LTM and STM Predictions
Final Prediction

Figure 3.1: The architecture of a multiple viewpoint system (adapted from Conklin & Witten, 1995; Pearce, 2005).

3.4.4 Combining Predictions from Finite Context Models

An important process in a multiple viewpoint systems is combining predictions in the form of probability distributions from a set of models, M, into a single probability distribution. Each model, $m \in M$ may be associated with a viewpoint m_τ, or be an

LTM or STM model. The probability of a viewpoint element, $t \in [\tau]$, from a model, m, given some context is simply denoted by $p_m(t)$.

The combination function itself must be monotonic, give a probability between the minimum and maximum probabilities from M, and the relative weightings for each m must be dependant on their certainty (Conklin, 1990, p. 70-71). Conklin (1990) introduces combining predictions with a weighted arithmetic mean:

$$p\left(t\right) = \frac{\sum_{m \in M} w_m p_m(t)}{\sum_{m \in M} w_m}. \tag{3.6}$$

Pearce et al. (2005) show a weighted geometric combination function (Equation 3.7) to be optimal for combining both viewpoint predictions and STM and LTM predictions when predicting `cpitch`. This combination function is upheld for the current research where R is a normalisation constant such that the entire distribution over $[\tau]$ sums to one.

$$p\left(t\right) = \frac{1}{R} \left(\prod_{m \in M} p_m\left(t\right)^{w_m} \right)^{\frac{1}{\sum_{m \in M} w_m}} \tag{3.7}$$

Shannon entropy, $H\left(p_m\right)$ (Equation 3.8) is used to quantify uncertainty, with more certain models gaining a higher weighting. The maximum entropy of a distribution is determined purely by the domain size, $|[\tau]|$ (Equation 3.9).

$$H\left(p_m\right) = -\sum_{t \in [\tau]} p_m(t) \log_2 p_m(t) \tag{3.8}$$

$$H_{max}\left(p_m\right) = \log_2 |[\tau]| \tag{3.9}$$

The weights themselves, w_m, are given by the relative entropy, $H_{relative}$ (Equation 3.10), of the distribution, p_m, of continuations for a given context.

$$H_{relative}\left(p_m\right) = \begin{cases} \frac{H(p_m)}{H_{max}(p_m)} & \text{if } H_{max}\left(p_m\right) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.10}$$

The weight w_m for a model is given by:

$$w_m = H_{relative}\left(p_m\right)^{-b}, \tag{3.11}$$

where $b \in \mathbb{Z}^*$ is a bias parameter giving an exponential bias towards models with lower relative entropy. When $b = 0$ the combination scheme is effectively unweighted. Note that if at any stage a viewpoint is undefined such that $\Psi\left(e_1^i\right) = \perp$ the viewpoint is removed from prediction in the combination process. If all viewpoints are undefined a uniform distribution over $[\tau_b]$ is returned.

3.4.5 Probability Distributions over ξ

m_τ will make predictions and return probability distributions over $[\tau]$. However, in order to make predictions of events (rather than merely viewpoint elements), these must be converted back to distributions over the basic event space, ξ, or more specifically, the domain of the basic attribute being predicted: $[\tau_b]$. For computational reasons (Conklin, 1990, p. 67-70) viewpoint predictions are carried out in stages for each τ_b in turn. For each τ_b a conversion function, Ψ'_τ, maps elements from $[\tau]$ onto any set of elements in $[\tau_b]$:

$$\Psi'_\tau : \xi^* \times [\tau] \rightarrow 2^{[\tau_b]} \tag{3.12}$$

where $2^{[\tau_b]}$ denotes the power set of $[\tau_b]$. Conklin and Witten (1995) and Pearce (2005) give no informed method for determining the order in which the viewpoints of interest, τ_{b^i}, are predicted since they predict only one viewpoint (`cpitch`). Whorley (2013, p. 115) elects to predict `Duration` followed by `Pitch` with no reason explicitly given, although it may be related to the fact that the domain size of `Duration` is far smaller than `Pitch`, and so is likely to have a lower cross-entropy. Interestingly, this is contradicted by the finding that, when electing which voice of a four-part chorale to predict first, it is found that predicting the voice with the largest domain and highest cross-entropy (the bass) first gives a lower cross-entropy for the whole system (Whorley et al., 2013a). Chapter 5 discusses in detail how multiple basic attributes can be predicted simultaneously as *merged attributes.*

3.5 Viewpoint Selection

The set of viewpoints that made up the early multiple viewpoint systems were hand selected, informed by intuition and background music theoretic knowledge (Conklin & Witten, 1995). A less arbitrary viewpoint selection method is proposed by Pearce (2005, p. 127-128) and a formalisation with slight modifications can be found in Whorley (2013, p. 126-136). An efficient search algorithm is necessary since most multiple viewpoint systems will have a large pool of primitive viewpoints available to them, which will increase exponentially when viewpoints are linked. Taking a pool of 14 primitive viewpoints, and allowing linked viewpoints between any two primitive viewpoints this gives a total pool of 105 (14 + 91) viewpoints. Therefore, the total search space will be equal to the size of the *power set* of viewpoints: $2^{105} \approx 4.1 \times 10^{31}$. This is a conservative estimate; in more advanced systems linking could occur between more than two viewpoints.

The viewpoint selection algorithm is a *forward stepwise selection* algorithm, starting with the set of basic viewpoints to be predicted, τ_{b^i}. The objective of the algorithm is to find the set of viewpoints which minimises the mean information content of the training corpus (Equation 3.5). Minimising the mean information content is, therefore, the heuristic for the algorithm, and is used to compare the performance of prospective models at each step. In brief, at each iteration, all single deletions and then additions to the given viewpoint set are considered. As smaller viewpoint systems are preferred over larger ones (following the principle of Ockham's razor), if a deletion step is selected the algorithm moves straight to the next iteration skipping the addition step, otherwise all additions are then trialled. The algorithm terminates when no deletions or additions will improve performance, guaranteeing a local (but not necessarily global) optimum.

3.6 Computational Complexity

The present research uses and builds on the IDyOM multiple viewpoint system implementation by Pearce (2005) in Common LISP.[10] The implementation stores viewpoint sequences as suffix trees (Bunton, 1996), constructed online with a generalised technique (Gusfield, 1997) capable of constructing a tree from multiple sequences, derived from the Ukkonen-Larson algorithm (Ukkonen, 1995). Therefore, given a training sequence of length n, and a query sequence of length j, a suffix tree constructed in $O(n)$ time and taking $O(n)$ space can be queried in $O(j)$ time. A single prediction run consisting of v viewpoints, each with a domain size $|[\tau]|$, over a single test sequence of length m must predict at every event a full probability distribution over the full domain of each viewpoint, giving a time complexity of $O(v \cdot |[\tau]| \cdot m^2)$, assuming that no order bound is placed on the PPM model.

[10] Pearce's (2005) implementation is open source and freely available for use: https://code.soundsoftware.ac.uk/projects/idyom-project.

Part II

Developments in Multiple Viewpoint Systems

Chapter 4

Musical Representation and Corpora

4.1 Overview

This chapter details the corpora and viewpoint representations used for the current thesis. Firstly, the notion of the *musical surface* and its relation to harmonic music and representation is introduced (§4.2). After a brief discussion weighing approaches to modelling harmony with multiple viewpoint systems (§4.3), the viewpoints representing harmonic and melodic properties are fully specified (§4.4). A discussion on representing temporal and metrical structure is given in §4.5. Finally, §4.6 describes the five datasets used in the thesis, alongside the preprocessing steps, and some basic statistical properties of the primary dataset.

4.2 Where is the Musical Surface in Harmonic Music?

In the current research, the *musical surface* is a methodological device defining the entry point of a computational model. More generally, the musical surface is broadly understood as a loosely defined phenomenon in the fields of music cognition, music perception, and computational musicology. A widely accepted understanding of the notion of a musical surface is a minimal, discrete representation of music into ordered atomic percepts. Often this will be at the note level (Lerdahl & Jackendoff, 1983; Sloboda, 1985; Wiggins, 2007), with the percepts being pitch and time (which may entail any of onset, duration, and offset). When building a computational model, or studying a cognitive phenomenon, the musical surface acts as the entry point of the model. Any processes which are not being studied are assumed to occur before the musical surface is formed, for example the perceptual process of identifying and categorising a single pitch from a complex sound wave. The processes which occur after the musical surface are the focus of the computational model or cognitive phenomenon in question. Lerdahl and Jackendoff's (1983) GTTM builds four levels of structure (grouping, metrical, time-space reduction, and prolongation reduction) from a musical surface comprising pitches and durations (essentially a piano-roll notation). Wiggins (2007) argues the case that the musical surface can be considered as the division between perceptual and cognitive processes, primarily beginning at the percepts of individual notes. However, Cambouropoulos (2010) argues that higher level cognitive processes such as beat tracking, metre induction, voice streaming, and chord identification (as exemplified by the task of automated music transcription) are required to define the musical surface. Multiple viewpoint systems (Conklin & Witten, 1995; Pearce, 2005) assume the musical surface to be composed of discrete events, comprising a number of features. These features form the basic attributes of the system, with derived and linked viewpoints analogous to cognitive processes working above the musical surface.

Despite the prevalence of note percepts (rather than chord percepts) in most descriptions of the musical surface, a harmonic musical surface is a perfectly tractable concept. Jackendoff (1987, p. 218) defines the musical surface as "encod[ing] the music as discrete pitch-events (notes and chords), each with a specific duration and pitch (or combination of pitches, if a chord)." However, perceptual and/or cognitive grouping processes are required to derive chords from simultaneous notes, which would deny the possibility of a harmonic musical surface following Wiggins (2007). Nevertheless, Cambouropoulos (2015) introduces two chord representation schemes very much in the flavour of Cambouropoulos (2010) to encode the harmonic musical surface: General Chord Type (GCT) and Directed Interval Class (DIC), which, respectively, must take as given the necessary cognitive functions required to identify tonal centres and categorise intervals.

The present research assumes a musical surface composed of chord symbols that occur at a fixed point in a metrical structure, and can be decomposed into chord roots and chord types. The process of grouping notes into chords, and categorising (according to a musical style) such groups into chord types is non-trivial and likely to require high-level cognitive processes. Taking a reductionist approach, these processes are not the focus of the current research, and therefore are assumed to be known at the entry point of the computational model under study. Chord symbols are a well-known attribute of music notation in lead sheets, and therefore are likely to hold some tractable relationship with similar cognitive percepts (Wiggins et al., 2010). Despite the use of music notation in building a representation scheme, the musical surface and lead sheet are not considered synonymous in the current research. Notably, a number of preprocessing steps (see §4.6.2) used to ensure simple, consistent percepts from lead sheet notation place the musical surface at a level abstracted from musical notation.

The distinction between the musical surface and musical notation is especially important when defining viewpoints that use tonal centres or keys, for example `cpintfref` (Pearce, 2005)[1] represents the interval between the current event and a referent pitch equivalent to the tonal centre. Although available in the musical score, the current research considers the identification of tonal centres to be on a cognitive level *above* the entry point of the model. In other words, tonal centres are identified by a listener with statistical models as higher-level cognitive processes (see Krumhansl, 1990; Temperley, 1999), in contrast to a performer or score reader where tonal centres are readily identified in the notation. Therefore, viewpoints requiring a tonal centre are avoided in the current research until a suitable scheme linking the statistical model and representation is developed (see Chapter 11). This research, therefore, defines the musical surface as an entry point for the phenomenon under study.

4.3 Representing Harmony with Multiple Viewpoint Systems

Multiple viewpoint systems take two distinct approaches when representing harmony. The first (Conklin, 2002; Whorley, 2013) represents harmony as the coincidence of polyphonic lines in a piece of music. The second (Conklin, 2010) represents harmony more abstractly by representing harmonic units as chord labels, similar to the music notation used for lead sheets (Pachet, Suzda & Martín, 2013), or the units of tonal harmonic musicological analysis.

The approach of Conklin (2002) and subsequently Whorley (2013) align coincident notes in multiple voices to find vertical and horizontal patterns in music with linked viewpoints across different voices (see §2.4.2). Such models are prone to sparsity issues given the size of their domains, which must be modified to balance the demands of time complexity whilst retaining a model that is able to generalise information (Whorley et al., 2013b). Although this method has the advantages of not requiring chord labelling and assuming minimal music theoretical knowledge, its modelling of harmony as a musicological construct is indirect. Polyphony is the immediate feature being modelled, any emergence of harmonic constructs are likely to be incidental as a result of strong

[1] Alternatively `ScaleDegree` in Whorley (2013), or `degree` in Conklin (2010).

correlations between voice leading and harmonic patterns in the corpus.

By contrast, Conklin (2010) models harmony more directly, outlining a multiple viewpoint representation for chord sequences of chord labels. Various viewpoints are defined, representing the scale degree and triad types of chords from their labels. In contrast to Conklin (2002) and Whorley (2013), chord voicing and inversion are not represented, allowing different voicings of the same chord to be considered as equivalent (as is generally accepted in music theory). Similarly, figurations and arpeggiations are no longer problematic (as they are for representations derived from Conklin, 2002) since the chord label is given directly.

This research follows the latter approach, taking the chord symbol as the musical surface and the entry level of the model. The chord labels from musicology and music notation are considered an approximate but meaningful proxy for cognitive constructs and representations of harmony.

4.4 Viewpoints

Viewpoints (see Chapter 3) are used to represent, and subsequently model, the multi-dimensional aspects of symbolic musical structure. Viewpoints, signified by a type, τ, are characterised by their partial function, Ψ_τ, mapping from sequences of events in the event space, ξ^*, to elements of the viewpoint domain, $[\tau]$. The following section details the partial functions of all primitive viewpoints used to represent both harmonic and melodic structure in the current research, summarised in Table 4-A for convenience.

4.4.1 Harmonic Viewpoints

This research takes the chord label as the entry point for modelling harmony (§4.3). A chord label comprises of four basic attributes: a root, and a chord type, a metrical position in a bar, and a bar length, defining the following event space:

$$\xi = [\texttt{Root}] \times [\texttt{ChordType}] \times [\texttt{PosInBar}] \times [\texttt{BarLength}]. \tag{4.1}$$

Each basic attribute is named by a basic type, τ_b, has an associated viewpoint where Ψ_{τ_b} is a projection function selecting the appropriate attribute from an event (Conklin, 1990), and a type set, $\langle\tau_b\rangle$, comprising of only the basic type itself.

Table 4-A: Summary of harmonic and melodic viewpoints used in the current research.

τ	$[[\cdot]]$	$[\tau]$	$\langle\tau\rangle$
`Root`	pitch class of root	$\{-1, 0, ..., 11\}$	`Root`
`ChordType`	chord type	see §4.4.1.1	`ChordType`
`PosInBar`	chord position in bar	$\mathbb{Z}^*$	`PosInBar`
`BarLength`	length of bar	$\mathbb{Z}^+$	`BarLength`
`RootInt`	sequential root intervals	$\{0, ..., 11\}$	`Root`
`RootIntFiP`	first in piece `RootInt`	$\{0, ..., 11\}$	`Root`
`RootInt` $\ominus$ `FiB`	threaded `RootInt`	$\{0, ..., 11\}$	`Root`
`MeeusInt`	Meeus root interval	$\{-2, -1, 0, 1\}$	`Root`
`MeeusIntFiP`	first in piece `MeeusInt`	$\{-2, -1, 0, 1\}$	`Root`
`ChromaDist`	cycle of fifth distance	$\{0, ..., 6\}$	`Root`
`ChromaDistFiP`	first in piece `ChromaDist`	$\{0, ..., 6\}$	`Root`
`MajType`	major triad	$\{major, minor, special\}$	`ChordType`
`7Type`	minor 7^{th}	$\{7, no7, NC\}$	`ChordType`
`FunctionType`	tonal function	see §4.4.1.3	`ChordType`
`FiB`	first in bar	$\{T, F\}$	`PosInBar`
`ICI`	inter-chord interval	$\mathbb{Z}^+$	`PosInBar`
`Pitch`	pitch of a note	$\mathbb{Z}$	`Pitch`
`Duration`	duration of a note	$\mathbb{Z}^+$	`Duration`
`Onset`	onset time of a note	$\mathbb{Z}^*$	`Onset`
`IOI`	inter-onset interval	$\mathbb{Z}^+$	`Onset`
`PitchInt`	sequential pitch intervals	$\mathbb{Z}^*$	`Pitch`

Note. Basic harmonic viewpoints (top section), viewpoints derived from `Root` (second section), derived from `ChordType` (third section), derived from `PosInBar` (fourth section), basic melodic viewpoints (fifth section), and derived melodic viewpoints (bottom section). Each is defined by their type τ, their semantic interpretation, $[[\cdot]]$, syntactic domain, $[\tau]$, and type set, $\langle\tau\rangle$.

4.4.1.1 Basic Harmonic Viewpoints

`Root.` `Root` is a pitch class denoted by the prefix of the chord label, for example the chord $B\flat^7$ has a root of $B\flat$. Informally, the root is the most important pitch of a chord; both perceptually and structurally. In western tonal harmony chords are considered as a stack of thirds (see Figure 4.1), with the bottom pitch of the stack representing the root of the chord. The root of the chord need not be the lowest sounding note, if the chord is inverted any note may be the lowest, acting as the bass, whilst the root of the chord remains the same.

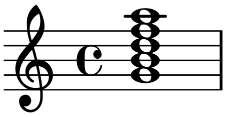


Figure 4.1: A G^{11} chord arranged as a stack of thirds, with the root, G, arranged at the bottom.

The present research assumes enharmonic equivalence, so that equivalent pitches with different pitch names are assigned the same symbol, for example, A♭ = G♯, E♭ = D♯, etc. Whilst pitch spelling can hold useful tonal harmonic information, this information is not necessarily available to the listener, for example in jazz the harmony is often provided by the piano which is tuned to equal temperament. An additional symbol is reserved to represent the *no chord* (*NC*) case.[2] *NC* is used on leadsheets to signify positions where no harmonic instruments are sounding (only melody notes are played), or occasionally for more free jazz styles any notes can be played. As there is no consistent meaning, the current research assigns it its own symbol. Therefore, the full domain[3] of `Root` can be defined as a set of 13 integers where -1 represents *NC* and 0 to 11 the remaining pitch classes:

$$[\texttt{Root}] = \{-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}.$$

The function[4] $[\![\cdot]\!]'_\tau : [\![\tau]\!] \rightarrow [\tau]$ maps between elements of the semantic domain of pitch classes and the `Root` domain. For clarity, the full mapping is given in Table 4-B.

`ChordType`. Other than the root, the acoustic quality of a chord is determined by its chord type, or the set of pitch classes that make up the chord. The chord type is usually signified by the suffix of the chord label, for example, the chord types of C^7, $F\sharp dim$, and Gm^7 are 7, *dim*, and *min7* respectively. After simplifying chord types (see §4.6.2) the full domain of `ChordType` is defined as follows:

$$[\texttt{ChordType}] = \{7, maj, maj6, min7, min, dim, halfdim, \\ min\sharp5, aug, alt, sus, special, NC\}.$$

Again, *NC* is used to indicate the *no chord* case which, by definition, always occurs when the `Root` is *NC*. An in depth discussion on the semantics of these chord types is

[2] A potential alternative would be to represent such *NC* occurrences with the undefined symbol, $\perp$. However, this would be undesirable since undefined symbols are filtered out by Φ_τ (Equation 3.2) when converting sequences of elements in the event space to sequences of elements in the viewpoint domain. This would lead to inadvertently learning sequences of non-adjacent elements.

[3] Assuming all elements are present in the test and training data.

[4] The inverse mapping of $[\![\cdot]\!]_\tau : [\tau] \rightarrow [\![\tau]\!]$, defined in §3.3.

Table 4-B: Semantic mapping between pitch classes and elements of the `Root` domain.

Pitch Classes		[`Root`]
NC	→	-1
B♯, C	→	0
C♯, D♭	→	1
D	→	2
D♯, E♭	→	3
E, F♭	→	4
E♯, F	→	5
F♯, G♭	→	6
G	→	7
G♯, A♭	→	8
A	→	9
A♯, B♭	→	10
B, C♭	→	11

Note. Only the most common pitch classes are shown, it is assumed that double accidental note names also map onto the corresponding `Root` element, e.g. D♭♭ → 0.

given in §4.6.2, and their typically associated pitch class sets given in Table 4-C. For the purposes of a viewpoint representation it is sufficient to state that they are atomic and unique.

Table 4-C: Prototype pcsets for chord types simplified by Algorithm 3.

Chord Type	Pitch Class Set
7	$\{0, 4, (7), 10\}$
maj	$\{0, 4, 7, (11)\}$
maj6	$\{0, 4, 7, 9\}$
min7	$\{0, 3, (7), 10\}$
min	$\{0, 3, 7\}$
dim	$\{0, 3, 6, (9)\}$
halfdim	$\{0, 3, 6, 10\}$
min♯5	$\{0, 3, 8\}$
aug	$\{0, 4, 8\}$
alt	$\{0, 4, 8, 10\}$
sus	$\{0, 5, 7, (10)\}$
special	$\{0, (1), (2), (5), (6), (8), (9), (10), (11)\}$
NC	$\{\}$

Note. Optional pitch classes are parenthesised.

`PosInBar` and `BarLength`. Metrical and temporal information is represented by `PosInBar`. A `PosInBar` element represents the number of temporal steps in a *timebase* unit the current event is from the start of the bar, where 0 is the first beat of the bar. `PosInBar` is often used as a derived viewpoint (Conklin & Witten, 1995; Pearce, 2005; Whorley, 2013), with `Onset` acting as the basic attribute for temporal information. This is not necessary for the current research where, by definition, a chord is present on the first beat of every bar (§4.6.2). Therefore, if necessary, it is possible to determine all chord onsets and durations from the metrical positions in the bar of a sequence of chords.

A timebase (Pearce, 2005, p. 63) is used to determine the granularity of the temporal representation. The timebase is an integer giving the number of time steps in a semibreve (four crochets).[5] A timebase of 8 is sufficient to represent all possible metrical positions in the harmonic corpora for the current research, so the smallest time step is a quaver.

Ordinarily, the domain of a basic viewpoint is determined purely from the corpus. However, this seems somewhat counter-intuitive for the domain of a viewpoint such as `PosInBar` as it would exclude the possibility of a chord occurring at perfectly plausible positions in the bar simply because it did not occur there in the training set. The present research, therefore, augments (c.f. Whorley et al., 2013b) the domain of `PosInBar` by pre-processing all chord durations in the dataset. The augmented domain is the set of all possible `PosInBar` elements that can be expressed as the sum of any number of durations, as well as 0 representing the first beat of the bar. During prediction the domain of `PosInBar` is set dynamically using `BarLength`. `BarLength` gives the length of the current bar in time steps and is not used as a predictive viewpoint in the present research. The domain of `PosInBar` at any given point is the set of non-negative integers greater than the `PosInBar` of the previous event, and less than the `BarLength` of the previous event. 0 is always added to the domain as it is possible at any time for the next chord to be on the first beat of the following bar.

4.4.1.2 Viewpoints Derived from `Root`

`RootInt`. Viewpoints that model the interval between successive events (e.g. `cpint`; Pearce, 2005) have been shown to be highly effective at modelling event attributes such as `cpitch` (e.g., Conklin & Witten, 1995; Pearce, 2005). The equivalent viewpoint for chord sequences is `RootInt` (Equation 4.2), the chromatic interval class between successive chord roots. Formally, `RootInt` is the difference in semitones between the current and previous events, modulo 12. For the first event of the sequence, where there

[5] A whole note, or four quarter notes, in American music theory.

is no previous event, the viewpoint element is undefined, and if either chord is an *NC* the viewpoint element is -1.

$$\Psi_{\texttt{RootInt}}(e_1^i) = \begin{cases} \perp & \text{if } i = 1 \\ -1 & \text{else if } \Psi_{\texttt{Root}}(e_1^i) = -1 \\ -1 & \text{else if } \Psi_{\texttt{Root}}(e_1^{i-1}) = -1 \\ \big(\Psi_{\texttt{Root}}(e_1^i) - \Psi_{\texttt{Root}}(e_1^{i-1})\big) \bmod 12 & \text{otherwise} \end{cases} \tag{4.2}$$

`RootIntFiP`. Some patterns arising from non-adjacent events can be captured with a class of viewpoint informally referred to as 'first in piece', signified by the suffix `FiP`. `RootIntFiP` (Equation 4.3) behaves similarly to `RootInt` but instead takes the root interval between the current root, and the first root of the piece. As a result, the first `RootIntFiP` of the piece is always 0. It is possible that `RootIntFiP` will provide some tonal structure, since the first chord of a piece is often the tonic or dominant. If this is the case then `RootIntFiP` may produce two clusters of statistical patterns around the scale degree (Riemann, 1895) of a chord (if the first chord is a tonic), or a scale degree displaced by seven semitones. Whilst it is not expected that this will result in a perfect tonal analysis of scale degrees for a given piece, it is expected that a similar statistical structure will emerge, albeit with a little noise.

$$\Psi_{\texttt{RootIntFiP}}(e_1^i) = \begin{cases} -1 & \text{if } \Psi_{\texttt{Root}}(e_1^i) \text{ or } \Psi_{\texttt{Root}}(e^1) = -1 \\ \big(\Psi_{\texttt{Root}}(e_1^i) - \Psi_{\texttt{Root}}(e^1)\big) \bmod 12 & \text{otherwise} \end{cases} \tag{4.3}$$

`RootInt` $\ominus$ `FiB`. The threaded viewpoint `RootInt` $\ominus$ `FiB` takes the root interval between chords on the first beats of successive bars. `FiB` is a test viewpoint (defined fully in §4.4.1.4) using a boolean to signify an event on the first beat of a bar. Like all threaded viewpoints `RootInt` $\ominus$ `FiB` returns undefined, ⊥, when `FiB` is false. The aim of the viewpoint is to capture some of the non-local, short-term structure of chord sequences. Note that since the inter onset interval between the first beat of successive bars will almost always be the same, and the current research does not use `RootInt` $\ominus$ `FiB` to predict `PosInBar`, the `ioi` element of the threaded viewpoint is omitted for this research[6].

[6] This follows the implementation by Pearce (2005) available at: https://code.soundsoftware.ac.uk/projects/idyom-project.

`MeeusInt`. The root progression music theories of Meeus (2000), Rameau (1971), and Schoenberg (1969) inspire the use of the `MeeusInt` viewpoint (Equation 4.4). Root progression theories describe tonal harmony exclusively through root transitions. Meeus (2000) simplifies all root progressions to a set of two: dominant progressions descend by a perfect fifth, descend by a third or ascend by a second, and subdominant progressions rise by a perfect fifth, ascend a third or descend a second. Conklin (2010) defines a similar type, `meeus`, although a full definition is not given (specifically, the value given to when `RootInt` = 6). In the current research, $[[1]]_{\texttt{MeeusInt}} = \textit{dominant}$, $[[-1]]_{\texttt{MeeusInt}} = \textit{subdominant}$, $[[-2]]_{\texttt{MeeusInt}} = \textit{tritone}$, $[[0]]_{\texttt{MeeusInt}}$ implies a repeated `Root`, and $[[-3]]_{\texttt{MeeusInt}}$ implies one or both events is a *NC*.

$$\Psi_{\texttt{MeeusInt}}(e_1^i) = \begin{cases} \perp & \text{if } i = 1 \\ 1 & \text{if } \Psi_{\texttt{RootInt}}(e_1^i) \in \{1,2,5,8,9\} \\ 0 & \text{else if } \Psi_{\texttt{RootInt}}(e_1^i) = 0 \\ -1 & \text{else if } \Psi_{\texttt{RootInt}}(e_1^i) \in \{3,4,7,10,11\} \\ -2 & \text{else if } \Psi_{\texttt{RootInt}}(e_1^i) = 6 \\ -3 & \text{otherwise} \end{cases} \tag{4.4}$$

`MeeusIntFiP`. `MeeusIntFiP` (Equation 4.5) relates to `MeeusInt` as `RootIntFiP` does to `RootInt`. Like `RootIntFiP`, `MeeusIntFiP` will always be defined 0 for the first event of the piece.

$$\Psi_{\texttt{MeeusIntFiP}}(e_1^i) = \begin{cases} 1 & \text{if } \Psi_{\texttt{RootIntFiP}}(e_1^i) \in \{1,2,5,8,9\} \\ 0 & \text{else if } \Psi_{\texttt{RootIntFiP}}(e_1^i) = 0 \\ -1 & \text{else if } \Psi_{\texttt{RootIntFiP}}(e_1^i) \in \{3,4,7,10,11\} \\ -2 & \text{else if } \Psi_{\texttt{RootIntFiP}}(e_1^i) = 6 \\ -3 & \text{otherwise} \end{cases} \tag{4.5}$$

`ChromaDist`. It is well established that tonal harmony progresses mainly in perfect fifths (seven semitones), reflected in pedagogical sources (Piston, 1948), functional theories (Riemann, 1895), root progression theories (Rameau, 1971) and chromatic pitch spaces (Longuet-Higgins, 1979). In general, root progressions which travel less distance on a cycle of fifths are preferred, for example, a progression of G to C is only one step on a cycle of fifths and is far more common than a progression of F♯ to C, which travels six steps. `ChromaDist` (Equation 4.6) acts on adjacent events, and is therefore undefined for the first event of the piece. The viewpoint uses the function CHROMA-DISTANCE (Algorithm 2, Appendix C) to return the chroma distance given a root interval, finding the minimal number of descending or ascending fifths required to complete the root interval.

$$\Psi_{\texttt{ChromaDist}}(e_1^i) = \begin{cases} \perp & \text{if } i = 1 \\ \textsc{Chroma-Distance}\left(\Psi_{RootInt}(e_1^i)\right) & \text{otherwise} \end{cases} \tag{4.6}$$

`ChromaDistFiP`. Finally, `ChromaDistFiP` (Equation 4.7) simply returns the `ChromaDist` between the current root and the first root of the piece. This models the tonal distance between the current chord and the first chord of the piece.

$$\Psi_{\texttt{ChromaDistFiP}}(e_1^i) = \begin{cases} \text{-1} & \begin{array}{l}\text{if } \Psi_{\texttt{Root}}(e_1^i) \text{ or} \\ \Psi_{\texttt{Root}}(e^1) = -1\end{array} \\ \textsc{Chroma-Distance}\left(\Psi_{RootIntFiP}(e_1^i)\right) & \text{otherwise} \end{cases} \tag{4.7}$$

4.4.1.3 Viewpoints Derived from `ChordType`

`MajType`. The symbolic nature of multiple viewpoint systems means that the chord type elements defined in §4.4.1.1 are considered to be equally different from one another. In other words, no pair of chord types are any more or less similar than any other pair. However, computational musicology (Chew, 2002; De Haas, Wiering & Veltkamp, 2013), music cognition (Krumhansl et al., 1982a), and jazz music theory (Levine, 1995) strongly suggest that this is not the case. Derived viewpoints such as `MajType` (Equation 4.8) group similar chord types together according to the properties of their pitch class sets (Table 4-C). `MajType` designates chords as *major* if they contain a note a major 3rd from the root (the pitch class set contains 4), *minor* if they contain a minor third (the pitch class set contains 3), and *special* if they contain no third or both major and minor thirds.

$$\Psi_{\texttt{MajType}}(e_1^i) = \begin{cases} \textit{major} & \text{if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{7, maj, maj6, aug, alt\} \\ \textit{minor} & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{min7, min, dim, halfdim, min\sharp5\} \\ \textit{special} & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{sus, special, NC\} \end{cases} \tag{4.8}$$

`7Type`. The seventh in a jazz chord gives a strong indication of the chord's function (Levine, 1989, 1995), with a minor seventh (a 10 in the pitch class set) implying a need for resolution, and a major seventh or no seventh giving a sense of closure. `7Type` (Equation

4.9) reflects this musicological phenomenon by separating chord types with and without a minor seventh, and reserves *NC* for the *no chord* case. A *sus* chord, whilst not necessarily containing a 10 in the pitch class set (see Table 4-C), is still categorised with the minor seventh chords as it requires resolution. In ambiguous cases (for example the *special* chord type), the chord type is grouped according to the distribution of unsimplified chord types before preprocessing (§4.6.2).

$$\Psi_{\texttt{7Type}}(e_1^i) = \begin{cases} 7 & \text{if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{7, min7, halfdim, alt, sus\} \\ no7 & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{maj, maj6, min, dim, min\sharp 5, aug, special\} \\ NC & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) = NC \end{cases} \tag{4.9}$$

`FunctionType`. Tonal harmony can be described functionally (Kostka & Payne, 1984; Piston, 1948) with chord types in jazz music giving a strong indication of the chord's functional properties. `FunctionType` (Equation 4.10) groups chord types into *tonic-major*, *tonic-minor*, *dominant*, and *pre-dominant*[7] categories. All chords with a major third and minor seventh are *dominant*, all other chords with a major third are *tonic-major*, all chords with a minor third and minor seventh are *pre-dominant*, all other minor chords are *tonic-minor*, and *NC* is retained for the *no chord* case. Note that chord type alone does not usually signify the function of a chord, context and the root can also play a substantial role. Nevertheless, grouping chord types by potential functional properties gives an indication of their similarity, and therefore may yield useful statistical patterns.

$$\Psi_{\texttt{FunctionType}}(e_1^i) = \begin{cases} tonic-major & \text{if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{maj, maj6, aug\} \\ tonic-minor & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{min, min\sharp 5\} \\ dominant & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{7, alt, sus, special\} \\ pre-dominant & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) \in \{min7, dim, halfdim\} \\ NC & \text{else if } \Psi_{\texttt{ChordType}}(e_1^i) = NC \end{cases} \tag{4.10}$$

4.4.1.4 Viewpoints Derived from `PosInBar`

`FiB`. A test viewpoint signifying with a boolean the first beat of a bar is given by `FiB` (Equation 4.11). The viewpoint is only used to define the threaded viewpoint `RootInt` $\ominus$

[7] A chord that precedes the dominant, typically *ii* or *IV*.

`FiB` (§4.4.1.2), at no point in the research is it used to predict `PosInBar`.

$$\Psi_{\texttt{FiB}}(e_1^i) = \begin{cases} T & \text{if } \Psi_{\texttt{PosInBar}}(e_1^i) = 0 \\ F & \text{otherwise} \end{cases} \tag{4.11}$$

`ICI`. Inter-chord interval (Equation 4.12), a derived viewpoint representing the temporal interval between successive chords, is equivalent to `ioi` in melodic viewpoint systems (Conklin & Witten, 1995; Pearce, 2005). The current research uses `PosInBar` and `BarLength` to calculate `ICI`.

$$\Psi_{\texttt{ICI}}(e_1^i) = \begin{cases} \perp & \text{if } i = 1 \\ \Psi_{\texttt{BarLength}(e_1^{i-1})} - \Psi_{\texttt{PosInBar}(e_1^{i-1})} & \text{else if } \Psi_{\texttt{PosInBar}(e_1^i)} = 0 \\ \Psi_{\texttt{PosInBar}(e_1^i)} - \Psi_{\texttt{PosInBar}(e_1^{i-1})} & \text{otherwise} \end{cases} \tag{4.12}$$

4.4.1.5 Linked Chord Viewpoints

In theory, a linked viewpoint may be formed from any number of constituent viewpoints (Conklin, 1990, pp. 61-63). However, in practice a limit is enforced on the number of constituent viewpoints.[8] For example, Pearce (2005) limits linked viewpoints to two constituent viewpoints. The current research imposes a limit of two constituent viewpoints in any linked viewpoint, or three if one of the constituents is `PosInBar`. This reduces the number of possible multiple viewpoint systems to a manageable amount for viewpoint selection, and ensures none of the domains for the linked viewpoints cause time complexity issues (§3.6), since the size of the domain of `PosInBar` is fixed to 1 as it is a given attribute (see §4.5). Any viewpoint that is not a linked viewpoint is referred to as a *primitive viewpoint.*

4.4.1.6 Sample Solution Array for Harmonic Viewpoints

A *solution array* (Conklin & Witten, 1995; Ebcīoğlu, 1986) is a structure used to align and store viewpoint sequences. For a basic event sequence e_1^N of N events represented by J primitive viewpoints a $N \times J$ matrix is produced. The location (i, j) takes the value

[8] Without a limit, the number of possible linked viewpoints for a multiple viewpoint system is equal to the cardinality of the powerset of all viewpoints. Assuming 12 viewpoints for the current research (ignoring `BarLength`) this gives 4096 possible linked viewpoints.

of $\Psi_{\tau_j}(e_1^i)$, which may be $\perp$ if $\Psi_{\tau_j}(e_1^i)$ is undefined. A sample solution array is given in Table 4-D.

Table 4-D: A solution array for the first four bars of *"Giant Steps"* by John Coltrane.

Type (τ)	Chord Symbols						
	B	D^7	G	$B\flat^7$	$E\flat$	Am^7	D^7
`Root`	11	2	7	10	3	9	2
`ChordType`	*maj*	7	*maj*	7	*maj*	*min7*	7
`PosInBar`	0	4	0	4	0	0	4
`BarLength`	8	8	8	8	8	8	8
`RootInt`	$\perp$	3	5	3	5	6	5
`RootIntFiP`	0	3	8	11	5	10	3
`RootInt` $\ominus$ `FiB`	$\perp$	$\perp$	8	$\perp$	8	6	$\perp$
`MeeusInt`	$\perp$	-1	1	-1	1	-2	1
`MeeusIntFiP`	0	-1	1	-1	1	-1	-1
`ChromaDist`	$\perp$	3	1	3	1	6	1
`ChromaDistFiP`	0	3	4	5	4	2	3
`MajType`	*maj.*	*maj.*	*maj.*	*maj.*	*maj.*	*min.*	*maj.*
`7Type`	*no7*	7	*no7*	7	*no7*	7	7
`FunctionType`	*tmaj.*	*dom.*	*tmaj.*	*tmaj.*	*tmaj.*	*pdom.*	*dom.*
`FiB`	*T*	*F*	*T*	*F*	*T*	*T*	*F*
`ICI`	$\perp$	4	4	4	4	8	4

Note. `FunctionType` elements of *tonic-major*, *dominant*, and *predominant* are respectively abbreviated to *tmaj.*, *dom.*, and *pdom.*

4.4.2 Melodic Viewpoints

The melodic corpora used at various points in the present research consist of melodic events (notes) comprised of three basic attributes: `Pitch`, `Duration`, and `Onset`, forming the event space:

$$\xi = [\texttt{Pitch}] \times [\texttt{Duration}] \times [\texttt{Onset}]. \tag{4.13}$$

`Pitch`. The `Pitch` attribute models the most prevalent percept of a note; the pitch height, which is a perceptual categorisation of the fundamental frequency of a sound. `Pitch` is exactly equivalent to `pitch` in Conklin and Witten (1995), `cpitch` in Pearce (2005), and `Pitch` in Whorley (2013). That is, pitch is an integer equivalent to a MIDI note number assuming enharmonic equivalence. As with all viewpoints, `Pitch` elements are considered equally different from one another, so, for example, 60 (C4) is considered equally different to 72 (C5) as it is to 54 (F♯3). The full theoretical domain of `Pitch` is the set of all integers, although in practice it will be the set of pitches seen in the test and training sets.

`Duration`. `Duration` measures the duration of a note from onset to offset in time steps set by the timebase. In previous multiple viewpoint research it is referred to as `duration` (Conklin & Witten, 1995), `dur` (Pearce, 2005), and `Duration` (Whorley, 2013). The full theoretical domain of `Duration` is the set of positive integers, which, in practice will be restricted to the set of durations seen in the dataset at hand.

`Onset`. `Onset` is a basic attribute representing the onset times of melodic events in timebase units, where 0 is the first beat of the first bar (even if the first bar is incomplete). Its definition and use is identical to `st` (start time) in Conklin and Witten (1995), `onset` in Pearce (2005), and `Onset` in Whorley (2013).

`PitchInt`. `PitchInt` (Equation 4.14) is a viewpoint derived from `Pitch`, returning the interval in semitones between the current and previous notes. In previous research it is referred to as `seqint` (Conklin & Witten, 1995), `cpint` (Pearce, 2005), and `Interval` (Whorley, 2013). Note that unlike `RootInt` (§4.4.1.2), `PitchInt` is the interval between pitches, and not pitch classes.

$$\Psi_{\texttt{PitchInt}}(e_1^i) = \begin{cases} \perp & \text{if } i = 1 \\ \Psi_{\texttt{Pitch}(e_1^i)} - \Psi_{\texttt{Pitch}(e_1^{i-1})} & \text{otherwise} \end{cases} \tag{4.14}$$

`IOI` The inter-onset interval (Equation 4.15) of a melodic event is the temporal interval between the current and previous events. This derived viewpoint is defined in an identical manner to previous research where is has been referred to as `gis221` (Conklin & Witten, 1995), `ioi` (Pearce, 2005), or `IOI` (Whorley, 2013).

$$\Psi_{\texttt{IOI}}(e_1^i) = \begin{cases} \perp & \text{if } i = 1 \\ \Psi_{\texttt{Onset}(e_1^i)} - \Psi_{\texttt{Onset}(e_1^{i-1})} & \text{otherwise} \end{cases} \tag{4.15}$$

4.5 Handling Temporal and Metrical Structure with Given Attributes

The majority of past research using multiple viewpoint systems as predictive models have focussed on predicting features such as note pitches rather than temporal features such as duration or onset. There are several notable exceptions: Pearce et al. (2010b) predicts inter-onset interval (IOI) and outer-onset interval (OOI) for segmentation, Conklin (2013b) uses a range of viewpoints derived from duration for folk song classification, and Whorley (2013) predicts duration in generating 4-part harmonised chorale melodies.

However, it is debatable whether sequences of metrical or temporal symbols necessarily hold truly Markovian properties. A typical sequence of durations might be $\mid 1, 1, 1, 1 \mid 1, 1, 2 \mid 4 \mid$ (where 1 is a crochet duration and $\mid$ indicate bar lines). A pure first-order Markov model would predict a 2 with equal probability after any event in the sequence. However, in order not to violate metrical structure (the sum of durations between each bar line is usually 4 unless there is syncopation), a duration of 2 is highly unlikely at the 4$^{\text{th}}$ position of the sequence, and a duration of 4 is highly unlikely in all but the 1$^{\text{st}}$, 5$^{\text{th}}$, and 8$^{\text{th}}$ positions. Variable order Markov models may, in part, improve prediction by taking more of the context into account, and multiple viewpoint systems can capture the metrical structure with a test viewpoint indicating the first beat of a bar. Nevertheless, hierarchical (Forth, 2012; Forth et al., 2016; Lerdahl & Jackendoff, 1983), or constrained-based (Roy & Pachet, 2013) approaches seem more natural and accurate computational approaches to temporal structure.

From a purely theoretical perspective, a relaxation on the treatment of domains must be employed for a multiple viewpoint framework to predict temporal information such as onset. Typically, `Onset` is treated as a basic attribute and, therefore, strictly would take its domain, [`Onset`], from the dataset at hand. However, this would produce a very large, or potential unbounded[9] viewpoint domain. In practice, the domain of `Onset` is bounded by setting it dynamically before each event prediction to be the set of possible onsets after adding all IOIs found in the corpus (see Conklin, 1990, pp. 84-85, and Pearce, 2005, p. 65). Note that this is necessary even when not predicting `Onset` directly, one case being when setting the domains of linked viewpoints containing `onset` or its derived viewpoints. Such an adjustment to the theoretical framework hints that temporal information should be considered as a special case.

The current research acknowledges the approach of Forth et al. (2016) by drawing a clear distinction between *what* is to be predicted and *when* it will happen, acknowledging that both tasks pose significantly complex problems in their own right. The primary focus of this work is the *'what'* component of symbolic prediction, with the *'when'* reserved for future research. Therefore, for most of the primary experiments of this thesis the prediction of temporal attributes, such as `PosInBar`, are removed from the predictive task. This is possible by introducing the notion of a *given attribute* to the multiple

[9] Depending on the interpretation of *"The domain of any basic attribute must contain all instances of the attribute that could be encountered in the chorales."* Conklin (1990, p. 85).

viewpoint framework. A given attribute is assumed to be known at the time of prediction, constraining the domain of the associated basic viewpoints and any derived viewpoints to the given attribute of the current event. In practice, this will only impact linked viewpoints containing the relevant[10] basic and derived viewpoints. Unless specifically stated, `PosInBar` is treated as a given attribute for the current research.

4.6 Corpora

As with all machine learning tasks, training and test data should be selected following a number of important criteria. In order to draw meaningful conclusion from models built on statistical induction the corpus must be structurally and stylistically coherent, and be large enough to produce non-sparse models that reflect significant regularities from the training data. From a practical perspective, the corpus must be relatively easy to obtain, be machine readable, and all of the required information be derivable from its original representation. Finally, in order for conclusions drawn from the research to have conviction, the corpus must be convincingly representative of the real-world data forming the subject of the study.

Further specific issues must be considered for a computational study of tonal harmony. As discussed in §4.3, this research models harmony as chord labels as opposed to vertical groupings of specific notes. Therefore, the chord labelling problem (the labelling of vertical grouping of notes or an audio segment with an appropriate harmonic symbol) is particularly relevant in the construction of the current corpus. Two general approaches are available. The first is to efficiently and reliably label an audio or staff notation corpus, or else use a corpus already labelled in this manner. However, this approach is fundamentally flawed as there are no chord labelling methods available which are practical, suitably accurate, and unbiased. Hand labelling a corpus of an appropriate size is often impractical, as well as being prone to human bias and error. There are currently no sufficiently accurate methods for automatic labelling, both from digital scores (at best 88% accuracy: Kröger, Passos, Sampaio & De Cidra, 2008) or from audio (ignoring methods prone to extreme overfitting, at best 82.85% accuracy: McVicar, Santos-Rodriguez, Ni & Bie, 2014). Furthermore, highly performing automatic labelling methods in both tasks (Mauch, Noland & Dixon, 2009; Pardo & Birmingham, 2002; Temperley, 2001) must often rely on musicological knowledge in the local context or global structure to inform chord label decisions. Although these techniques are perfectly sound for music information retrieval (MIR), they create a positive feedback bias in modelling tasks, where the

[10] If the viewpoint's typeset contains the basic viewpoint associated with the given attribute.

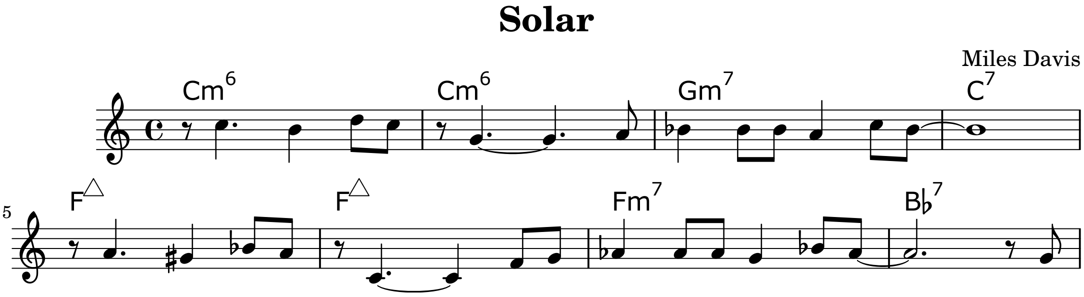


Figure 4.2: Extract from a leadsheet: *"Solar"* by Miles Davis

goal is to assess the underlying structure of sequences. For example, Temperley's (2001) third Harmonic Preference Rule prefers (in ambiguous cases) adjacent roots to be close on the line of fifths, meaning that any model built from the labelling will have a bias towards root progressions by fifths.

The second approach is to avoid the labelling problem entirely. This is achieved by choosing a corpus where labels are explicitly written by the composer, for example, jazz or pop leadsheets. A leadsheet (see Figure 4.2) is a score containing the main melody in stave notation, the chord sequence above as chord symbols and, where appropriate, lyrics below the stave. The leadsheet represents precisely all of the information of a piece which remains invariant between performances, therefore the corpus is built on representations of compositions directly, and not representations of specific performances.

4.6.1 Datasets

Five datasets are used as testing and training sets in the present research, summarised in Table 4-E. The primary dataset (1) consists of the chord sequences of jazz standards from *The Real Book Vol. 1* (Leonard, 2012), compiled by Pachet et al. (2013).[11] Historically, well known jazz pieces (known as *jazz standards*) were notated and collected in *fake books*, used as both a performance and a learning aid (Witmer & Kernfeld, 2002). *The Real Book* is one such collection compiled in the 1970's in the Boston area, likely by various young musicians from Berklee College of Music. Although popular among professional and student jazz musicians, most fake books (including the original *Real Book*) breached copyright laws, and therefore were largely distributed informally through word of mouth, making their precise origins difficult to trace. Most fake books were plagued with inaccuracies and inconsistencies, however, the popularity of *The Real Book* was in part due to its (relatively) accurate transcriptions. The original source contains 444 leadsheets, however, preprocessing steps (see §4.6.2) remove leadsheets with ambiguous

[11] Available to view at http://lsdb.flow-machines.com.

structures or section orders leaving 348 reliable compositions. The dataset does not contain reliable key signature or tonal centre information, so this is not used for the current research.

Four further datasets are used throughout the thesis to test if various results are specific to a corpus, or more general. A second harmonic dataset (2) comprises the chord sequences from all 179 Beatles songs, compiled by Harte, Sandler, Abdallah and Gómez (2005). The three melodic datasets, all used by Pearce and Wiggins (2004), are a set of 185 Bach chorale melodies from Riemenschneider (1941), 556 German folksongs from the Essen Songbook Collection (Schaffrath, 1995), and 152 Canadian folksongs ballads from Nova Scotia (Creighton, 1966). The basic attributes of the harmonic datasets are `Root`, `ChordType`, and `PosInBar`, whilst the melodic dataset uses basic attributes of `Pitch` and `Duration` only.

Table 4-E: Two harmonic and three melodic datasets used in the current research.

ID	Description	Pieces	Events	Basic Attributes	Timebase
1	Real Book Vol. 1	348	15,197	`Root`, `ChordType`, `PosInBar`, `BarLength`	8
2	Complete Beatles	179	17,557	`Root`, `ChordType`, `PosInBar`, `BarLength`	8
3	Bach chorales	185	9,227	`Pitch`, `Duration`, `Onset`	96
4	German folksongs	566	33,087	`Pitch`, `Duration`, `Onset`	96
5	Canadian folksongs	152	8,552	`Pitch`, `Duration`, `Onset`	96

4.6.2 Preprocessing Steps

A few preprocessing steps are necessary in order to make the corpora consistent and reliable. Note that these preprocessing steps have methodological implications in creating a distinction between the entry point of the model, considered to be the musical surface of interest in the current research (see §4.2), and the musical notation of the corpora. In the harmonic datasets, a pre-processing step is used to reduce the domain size of the original `ChordType` attribute and to incorporate slash chord notation.[12] Each chord is

[12] Where the bass note of a chord is stated explicitly after a backslash, e.g G^7/C. Note the bass note need not be present in the original chord type, in the example given C is the bass note even though it

converted to a *pcset* (see Forte, 1973) and transposed so that the root (signified by the chord symbol prefix) is 0. In the case of slash chords, the bass note (the note name after the slash) is considered to be the root unless it is already present in the chord, or if it is a minor or major 7th (10 or 11 semitones) above the root of the chord. This would imply that the bass note of a slash chord signifies only a change of inversion, but not function, (Levine, 1995, Ch. 5). Finally, Algorithm 3 (see Appendix C) assigns one of 13 symbols to a given pcset according to the combination of pitch classes in the set. Functionally equivalent chords with differing notation can, therefore, be represented by the same symbol. For example, $C^{11}(no3rd)$ and G^7/C both represent 11th chords with an omitted major 3rd and a functional root of C. This also allows large `ChordType` viewpoint domains to be reduced from 67 and 64 chord types for datasets 1 and 2 respectively. Large domains can greatly reduce the speed of a multiple viewpoint system (see §3.6, and Whorley et al., 2013b) and so it is also advantageous to reduce them for practical reasons. All 13 `ChordTypes`, including the special symbol *NC* (signifying that no chord is being played), are given in Table 4-F alongside corresponding typical chords from the *Real Book Vol. 1* source.

Table 4-F: The complete alphabet of `ChordType` with typical corresponding chords mapped from the *Real Book Vol. 1* using Algorithm 3.

`ChordType`	Chord types from the *Real Book Vol. 1*
7^{th}	7, 9, 13, $7\sharp9$, $7\flat9\flat5$
$maj.$	M, M^7, M^9, $M^{7\flat9}$, $M^{7\flat5}$
$maj.6^{th}$	6, $6^{\sharp}11$, 69
$min.7^{th}$	m^7, m^9, m^{13}, $m^{7\sharp5}, m^{7add4}$
$min.$	m, m^6, m^{69}, m^{M7}, m^{add9}
$min.\sharp5$	$m^{\flat6}$, $m^{\sharp5}$
$dim.$	dim, $dim7$, $m^{\flat5\flat13}$
$halfdim.$	$halfdim7, m^{7\flat5\flat13}, m^{7(\flat5\flat2)}$, $m^{7\flat5\sharp5}$
$aug.$	$+$, $aug\sharp4$
$alt.$	$7\sharp5$, $9\sharp5$, $13\sharp9\sharp5$, $7\sharp5\flat5$
sus	$7sus$, $sus2$, $6sus4$, $13\flat9sus$, $Phrygian$
$special$	various unclassified slash chords e.g. $FM^7/\ E\flat$
NC	NC

There is a notable and significant difference between a notated leadsheet and how it is both performed and experienced aurally. A typical notated leadsheet will only contain one or two notated sections repeated an indeterminate number of times in performance,

is not in a G^7 chord.

often with a different melody in solo sections but importantly retaining the same chord sequences. This poses a question to the nature of the training data for machine learning algorithms, which is: what is it that is being learned? A machine learning algorithm tasked with capturing the features of a composer might be inclined to ignore repeats, since the chord sequence is only written and composed once. On the other hand, machine learning algorithms aiming for perceptual and cognitive models of human perception might be more inclined to align themselves with the perspective of the listener and include multiple repeated sections in the training data. The current research takes this approach, with slight compromises as detailed below.

Where a fixed number of repeats are specified for a section, an 'unfolded' version of the leadsheet with the correct number of repeats is created so that chord sequences spanning the start and end of sections are correctly accounted for. Some leadsheets indicate an 'open' repeat which may repeated any number of times, although is assumed to be a single repeat for the purposes of the current research. Other leadsheets contain repeat structures that are unspecified or open to the performer's interpretation. In order to avoid inadvertently learning false chord sequences that span various sections not intended to be performed sequentially, these leadsheets are identified and removed from the training corpus. Additionally, leadsheets that are comprised purely of *NC* chords (i.e. they are completely melodic) are also removed.

The repeated nature of many leadsheets, coupled with the brevity of notational conventions in jazz mean that often the final notated chord of a leadsheet will be a dominant chord (V^7) leading back to the first chord of the leadsheet. In practice, on the final repeat of the section the final chord would be resolved in a manner agreeable to the performers (some form of tonic chord), although this is not explicitly notated. This presents a potential problem, as a large number of leadsheets in the training data do not end with the final chord as performed or heard by the listener. To resolve this a preprocessing step recognises when the final notated chord resolves onto the first chord of the leadsheet, and simply appends the first chord to the end of the leadsheet. The first chord is considered to act as a resolution for the final chord if it is a perfect 5th below the final chord (or the `RootInt` = 7), and if the `ChordType` of the final chord contains a major 3rd and minor 7th (4 and 10 are members of the pcset).

As a notational source, leadsheets were originally transcribed informally by a variety of people, and therefore, can lack notational consistency. In particular, there is no convention for when a repeated chord should be re-written; usually a blank bar will indicate to repeat the previous chord or sometimes the chord will be re-stated explicitly. To enforce consistency, the first beat of any bar is always deemed to have a chord by definition, if it is not stated it takes the `Root` and `ChordType` from the previous chord.

Chords repeated (with the same `Root` and preprocessed `ChordType`) within a bar are removed, and their duration added to the previous chord. Often chords repeated within a bar in the leadsheet indicate a repeated figuration, although this use of notation is inconsistent, and figurations or harmonic realisations are not the focus of the current research. However, chords may still be repeated between bars, retaining the notion of a Markov transition which returns to the same state which would be otherwise be lost if all repeated chords were to be removed.

4.6.3 Basic Statistical Properties of the *Real Book Vol. 1* Dataset

Some cursory basic statistical properties of the primary dataset (*Real Book Vol. 1*) are useful at this stage to give a flavour for the training data. The dataset of 348 leadsheets contains 15,197 events, with a mean of 43.70 events per leadsheet, and a standard deviation of $\sigma = 20.42$. The fewest number of chords in a leadsheet is 6, and the maximum 155. The relatively high standard deviation and range exemplify the variety of leadsheets in the database, with some being a sketch of only a handful of chords, and others being almost fully written-out compositions.

The zero-order distributions of the three basic attributes of `Root`, `ChordType`, and `PosInBar` are given in Figure 4.3, Figure 4.4, and Figure 4.5 respectively. The `Root` distribution is reminiscent of Krumhansl's (1990) pitch class profiles, and shows C, D, F, and G as the most common chord roots. The non-uniform property of the distribution suggests that the underlying distributions of key signatures and tonal centres is also non-uniform. The `ChordType` distribution strongly favours *7, maj,* and *min7* chord types (78.0% of all chords), with *aug, min♯5, special* and *NC* being extremely rare (1.5%) in comparison. This minimizes any ambiguity in the simplified chord types (Table 4-F), as the chord types prone to ambiguity are extremely rare. Finally, the `PosInBar` distribution shows that over three quarters of chords fall on the first beat of the bar, with most of the rest falling on the third (assuming four crochet beats in a bar).

4.7 Summary

This chapter has provided the representational foundations of the thesis in terms of viewpoints (§4.4) and presented the corpora to be used in the empirical research of the proceeding chapters (§4.6). In addition, related discussions concerning the definition of a musical surface of symbolic harmonic music (§4.2), the representation of harmony as chord symbols with multiple viewpoint systems (§4.3), and the handling of temporal and

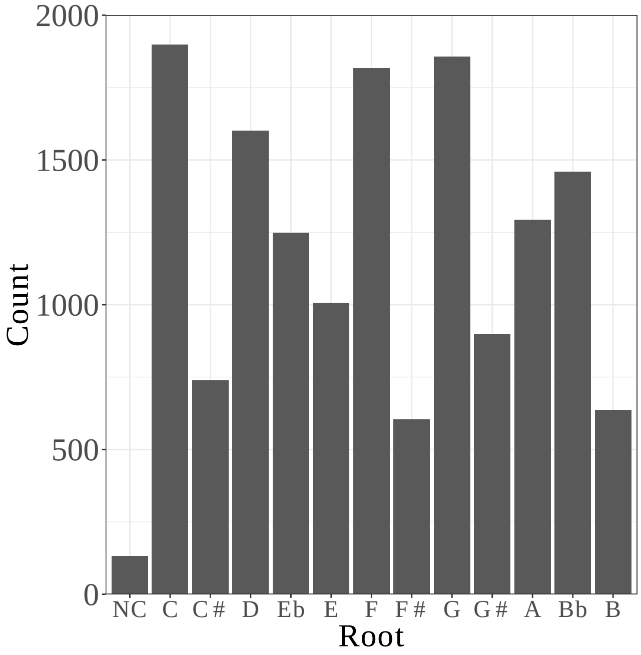


Figure 4.3: Zero-order distribution of `Root` elements in dataset 1.

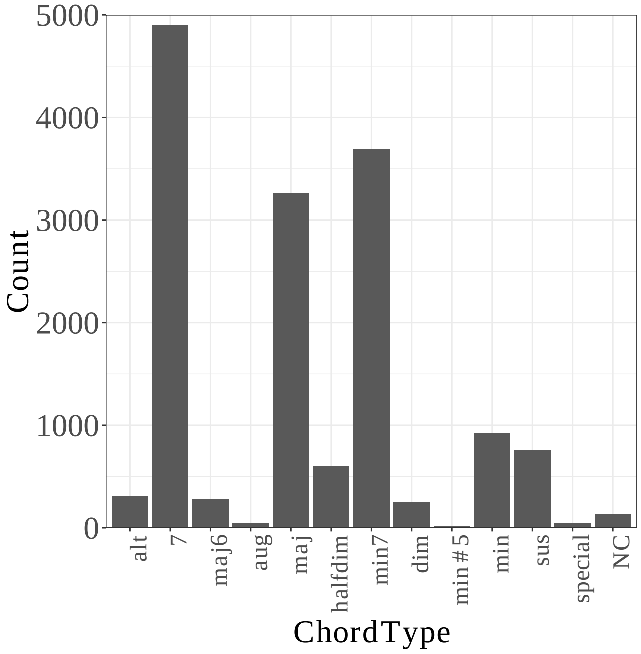


Figure 4.4: Zero-order distribution of `ChordType` elements in dataset 1.

metrical structure in relation to Markovian prediction (§4.5) have been presented.

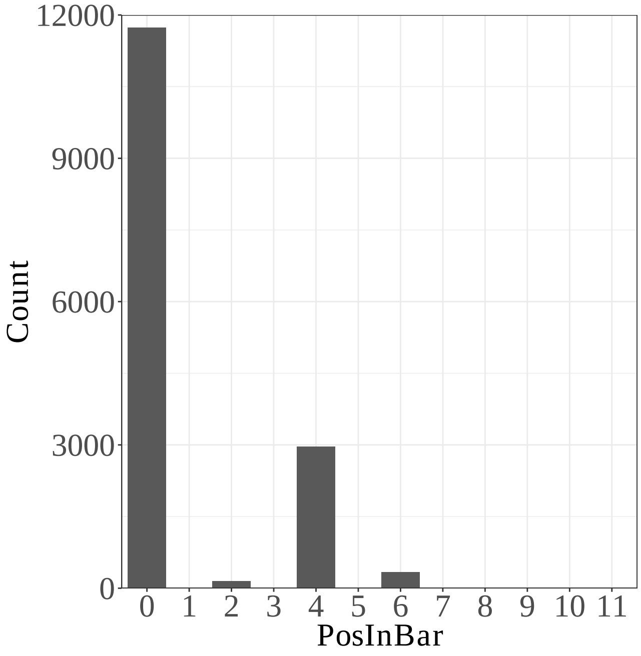


Figure 4.5: Zero-order distribution of `PosInBar` elements in dataset 1 in timebase units (1 = quaver). Note that element 3 has a count of 1, and element 10 a count of 2.

Chapter 5

Predicting Merged Attributes with Multiple Viewpoint Systems

5.1 Overview

The domain of chord sequence prediction with multiple viewpoint systems is relatively understudied in comparison with melodic prediction at the note level. This chapter aims to significantly contribute to the understanding of how multiple viewpoint systems predict chord sequences. A key issue is in addressing the level of representation of chord symbols (discussed in §4.3); either dividing them into two percepts of `Root` and `ChordType`, or considering them as the single percept of `Root`$\otimes$`ChordType`. This chapter proposes and tests a position between theses two stances, whereby chord symbols can be matched as a single percept, but viewpoints can enact on one or other percept independently. Such percepts are referred to as *merged attributes*, formally introduced in §5.5, before being tested in experiments that predict merged attributes with individual (§5.6) and multiple (§5.7) viewpoints. A second issue is finding the optimal *smoothing techniques* (formally introduced in §5.2) when applying statistical learning algorithms to novel datasets and domains. In order to ensure that future experiments of the current thesis are not a result of unusual smoothing artefacts, this is addressed first in §5.4.

5.2 Statistical Learning of Sequential Musical Data

Statistical, and especially Markovian, approaches to modelling sequential data frequently encounter two problems when employing fixed-order models. Firstly, symbols that are novel to the context may be encountered, resulting in probabilities of zero being given to new events. This has been identified by Witten and Bell (1991) as the *zero-frequency problem*. Secondly, it is difficult to determine the context length (or model order) that will give the best predictions. In general, longer contexts can give more specific predictions, however, they are more likely to encounter sparsity issues than shorter contexts. Many approaches address this by combining models with different context lengths (Ron et al., 1996), a naive implementation of which would result in polynomial time and space algorithms. A large number of proven methods and techniques, collectively referred to as *smoothing techniques*, have been established to tackle these problems. The techniques established in the task of predicting musical sequences (Conklin & Witten, 1995; Pearce & Wiggins, 2004), to be tested in the current research (§5.4) are presented in §5.2.1. Alternative approaches and techniques are summarised in §5.2.2.

5.2.1 PPM and Smoothing Techniques used in IDyOM

Prediction by Partial Match (PPM) uses a collection of techniques from data compression known as *smoothing techniques* which improve the performance of Markov models, in particular tackling problems associated with zero-frequency counts (Witten & Bell, 1991) and fixed order models. In general, this is achieved by adjusting the maximum likelihood estimates to save probability mass for novel events, and finding a way to combine models of different orders in a meaningful way. Pearce and Wiggins (2004) test a number of these techniques, which are later implemented in the IDyOM model (Pearce & Wiggins, 2012).

Two frameworks exist to achieve this aim, commonly referred to as *backoff smoothing* (Kneser & Ney, 1995) and *interpolated smoothing* (Chen & Goodman, 1999; Jelinek & Mercer, 1980). Both frameworks utilise a global order bound, g, to recursively combine predictions from the g^{th} order down to the -1^{th} order. $\alpha(e_i \mid e_{i-n+1}^{i-1})$ represents the *prediction probability*, essentially an adjusted maximum likelihood estimate (Equation 3.3). $\gamma(e_{i-n+1}^{i-1})$ is the *escape probability*, or the amount of weight given to lower order models. $t(e_i^j)$ is the *type count* of a sequence and returns the number of different symbol types seen after the sequence e_i^j. $t(\varepsilon)$ is the type count of the empty sequence; in other words, the total number of symbol types already seen by the model. The fundamental difference between backoff and interpolated smoothing is that backoff smoothing (Equation 5.1) escapes to the next order only when it encounters a novel symbol for a given context, whilst interpolated smoothing (Equation 5.2) always escapes to the lower order, blending predictions from the $(n-1)^{th}$ and $(n-2)^{th}$ orders recursively until termination after the 0^{th} order (when $n = 1$).

$$p\big(e^i \mid e_{i-n+1}^{i-1}\big) = \begin{cases} \dfrac{1}{\mid \xi \mid +1 - t(\varepsilon)} & \text{if } n < 1 \\ \alpha\big(e^i \mid e_{i-n+1}^{i-1}\big) & \text{if } c\big(e^i \mid e_{i-n+1}^{i-1}\big) > 0 \\ \gamma\big(e_{i-n+1}^{i-1}\big) \cdot p\big(e^i \mid e_{i-n+2}^{i-1}\big) & \text{otherwise} \end{cases} \tag{5.1}$$

$$p\big(e^i \mid e_{i-n+1}^{i-1}\big) = \begin{cases} \dfrac{1}{\mid \xi \mid +1 - t(\varepsilon)} & \text{if } n < 1 \\ \alpha\big(e^i \mid e_{i-n+1}^{i-1}\big) + \gamma\big(e_{i-n+1}^{i-1}\big) \cdot p\big(e^i \mid e_{i-n+2}^{i-1}\big) & \text{otherwise} \end{cases} \tag{5.2}$$

5.2.1.1 Unbounded Length Contexts

Many of these smoothing techniques were developed as variants of PPM, a leading data compression scheme. The original algorithm proposed by Cleary and Witten (1984) used backoff smoothing with a fixed order bound and made use of two escape methods: A and B (see §5.2.1.2). Variations of PPM relevant to the current research include additional escape methods (Howard, 1993; Moffat, Neal & Witten, 1998; Moffat, 1990), the use of interpolated smoothing (Bunton, 1997), and *unbounded length contexts* (Cleary & Teahan, 1997), also known as PPM*. Unbounded length contexts remove the need for a fixed order bound by exploiting the fact that novel symbols tend to occur less frequently after *deterministic* contexts (ones which are followed by only one symbol type, i.e. $t(e_i^j) = 1$) than *non-deterministic* contexts (where $t(e_i^j) > 1$) when compared to a uniform prior distribution (Cleary & Teahan, 1995). Cleary and Teahan (1997) propose that for unbounded length contexts an order bound can be found dynamically for each symbol in a sequence by selecting the shortest deterministic context as the order bound, or, if no such context exists, the longest matching context.

5.2.1.2 Escape Methods

Escape methods are different methods for calculating $\alpha(e^i|e_{i-n+1}^{i-1})$ and $\gamma(e_{i-n+1}^{i-1})$, determining the amount of weight assigned to novel events for a given context. Table 5-A summarises five escape methods reviewed and empirically tested by Pearce and Wiggins (2004). Method A (Cleary & Witten, 1984) effectively assigns a count of one to all novel events given a context. Method B (Cleary & Witten, 1984) introduces the type count $t(e_{i-n+1}^{i-1})$ to the escape probability, so that more weight is given to novel symbols occurring after a context that is usually followed by more symbol types. Additionally, the effect of anomalies is reduced by subtracting one from the symbol count of the prediction probability so that novel symbols must occur twice before they are counted. Moffat (1990) proposed method C (also known as *Witten-Bell smoothing*) as a hybrid of the previous two methods, with the escape probability adjusted by the type count as in method B, but the symbol count of the prediction probability unaltered as in method A. However, forcing symbols to occur twice before they are counted by subtracting one from the prediction probability is considered wasteful. Howard (1993) proposes method D as a compromise, which subtracts only half from the symbol counts of the prediction probability. Finally, method AX (Moffat et al., 1998) is a simplified and corrected version of method P (Witten & Bell, 1991), which assumes the occurrence of novel events follows a Poisson distribution. Note the special type count $t_1(e_{i-n+1}^{i-1})$ signifies the number of symbol types which have appeared only once after a given context. In data compression studies, methods similar to method AX tend to outperform methods C and D, with methods A and B performing worst (Bunton, 1997; Cleary & Teahan, 1997; Moffat, Sharman, Witten & Bell, 1994; Witten & Bell, 1991). Since various qualities of the training and test data, such as alphabet size and skew (Moffat et al., 1994), have an impact on the performance of different escape methods, there is no informed way of selecting an escape method without *a priori* knowledge of the corpus (Witten & Bell, 1991). The optimal escape method can, therefore, only be found with an experimental approach.

Table 5-A: Prediction and escape probabilities of five escape methods empirically tested by Pearce and Wiggins (2004).

Escape Method	Prediction Probability $\alpha\big(e^i \vert e_{i-n+1}^{i-1}\big)$	Escape Probability $\gamma\big(e_{i-n+1}^{i-1}\big)$
A	$\frac{c\left(e^i\vert e_{i-n+1}^{i-1}\right)}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)+1}$	$\frac{1}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)+1}$
B	$\frac{c\left(e^i\vert e_{i-n+1}^{i-1}\right)-1}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)}$	$\frac{t\left(e_{i-n+1}^{i-1}\right)}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)}$
C	$\frac{c\left(e^i\vert e_{i-n+1}^{i-1}\right)}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)+t\left(e_{i-n+1}^{i-1}\right)}$	$\frac{t\left(e_{i-n+1}^{i-1}\right)}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)+t\left(e_{i-n+1}^{i-1}\right)}$
D	$\frac{c\left(e^i\vert e_{i-n+1}^{i-1}\right)-0.5}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)}$	$\frac{0.5\cdot t\left(e_{i-n+1}^{i-1}\right)}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)}$
AX	$\frac{c\left(e^i\vert e_{i-n+1}^{i-1}\right)}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)+t_1\left(e_{i-n+1}^{i-1}\right)+1}$	$\frac{t_1\left(e_{i-n+1}^{i-1}\right)+1}{\sum_{e\in[\tau]} c\left(e\vert e_{i-n+1}^{i-1}\right)+t_1\left(e_{i-n+1}^{i-1}\right)+1}$

5.2.1.3 Update Exclusion

Update exclusion is a method proposed by Cleary and Witten (1984) that aims to improve probability estimates with an altered counting scheme for n-grams. The rationale behind the method is that, when escaping down to lower orders, n-grams which would have already been seen at higher orders (therefore preventing escape in the case of backoff smoothing) are still included in calculating predictions for the lower order models. This wastes a portion of the probability mass which would otherwise be assigned to possible predictions. To borrow an example from Cleary and Teahan (1997), the task is to give a probability estimate to the symbol *'d'* following the sequence *'abracadabra'* with a backoff model, with an order bound of $g = 2$. A 3-gram model will escape to the lower order, since *'d'* does not occur in the context of *'ra'* in the sequence. Without update exclusion, a simple maximum likelihood estimate would assign a probability of $\frac{1}{4}$ for the 2-gram model, as the context *'a'* occurs four times, and is followed by a *'d'* on one of those occasions. If update exclusion is used, the 2-gram model will give a maximum likelihood estimate of $\frac{1}{3}$, since *'c'* has already been seen in the context of *'ra'* and is therefore removed from the predictions following the context *'a'*.

5.2.1.4 IDyOM Smoothing Parameters

Pearce and Wiggins (2004) established empirically the optimal smoothing parameters for IDyOM on a variety of monophonic melodic datasets. Interpolated smoothing consistently outperformed backoff smoothing across all datasets and for most other smoothing parameter combinations. Method C was the most consistently high performing escape method, although method AX also performed well for the STM. The effect of update exclusion was found to be sensitive to other smoothing parameters, the dataset, and whether the LTM or STM was being used. Unbounded length contexts (PPM*) outperformed the best fixed-order models when combined with interpolated smoothing, but with backoff smoothing, improvements were inconsistent. The LTM+ was found to outperform the STM when both were given the best smoothing parameters. Overall, the best LTM+ and STM were both unbounded, used interpolated smoothing and escape method C, but not update exclusion.

Models are given a shorthand notation (Pearce & Wiggins, 2004) indicating their set of smoothing parameters. The long- and short-term models are depicted by LTM and STM respectively, with LTM+ representing the hybrid model (see §3.4.3). Escape methods are indicated by 'A', 'B', 'C', 'D', and 'X' (for method AX). The order bound is given by an integer, or '*' if unbounded. If update-exclusion is used, a 'U' appears next in the shorthand string. An ‘I’ shows the model uses interpolated smoothing, otherwise backoff smoothing is used. For example, the best performing models found by Pearce and Wiggins (2004) were LTM+C*I, a hybrid long-term interpolated smoothing model, using escape method C and unbounded length contexts, and STMC*I, a short-term model otherwise with the same parameters.

5.2.2 Related Statistical Approaches

Several related string matching and statistical learning approaches have been established in the literature, with performance compared in terms of data compression (equivalent to mean information content, see §3.4.2). The well-known lossless data compression algorithm by Ziv and Lempel (1978), applicable to sequence prediction (Rissanen, 1983), parses left-to-right adding unique phrases to a dictionary used to construct a prediction tree. Encountering symbols novel to a given context (which may be empty) triggers a return to the root of the tree. A Prediction Suffix Tree (PST) (Bejerano & Yona, 2001; Ron et al., 1996) forms a suffix set consisting of all substrings in the training set not exceeding an order bound and occurring sufficiently frequently. Additionally, given the prediction of a symbol, suffixes are retained only if their maximum likelihood estimate is larger (defined by a hand-coded parameter) than the corresponding parent suffix which is one symbol shorter. Context Tree Weighting (CTW) models (Willems, Shtarkov & Tjalkens, 1995) combines predictions from all suffix trees within a bounded depth with probability estimates calculated using a Krichevsky-Trofimov estimator (Krichevsky & Trofimov, 1981) with a computationally efficient recursive process. Originally implemented for binary alphabets, Volf (2002) proposes an extension to finite alphabets of arbitrary size with a hierarchical decomposition into binary decisions. In terms of data compression performance metrics, Begleiter et al. (2004) provides a comprehensive comparison of the Lempel-Ziv, CTW, PST, and PPM algorithms over English text (the Calgary copurs), music in MIDI format, and protein sequences. CTW and PPM were found to perform best with an average log-loss of 3.02 / 3.03 bits/symbol for English text, 1.21/1.30 bits/symbol for MIDI files and 4.56/4.48 bits/symbol for protein sequences.

Similar methods have been developed as the underlying mechanism behind symbolic music generation systems capable of composing or improvising in any style given an appropriate training corpus. For modelling musical style and generation, Dubnov, Assayag, Lartillot and Bejerano (2003) compares Lempel-Ziv and PST methods, finding that Lempel-Ziv is able to run in real time and find musically coherent motifs. However, it is also prone to replicating large sequences of the training data joined with unexpected juxtapositions. Conversely, PST creates interesting transitions between replicated sequences, although may produce out of style notes and cannot be run in real time. Pachet (2003) presents an interactive music generation system built on prefix trees which learns all substrings of the training corpus and runs in real time. Musical generations are composed through a random walk method, falling back to shorter contexts in the prefix tree if the relevant context has not been seen in the training data. The system is reported to be able to produce fast tempo jazz improvisations which are stylistically indistinguishable from the user's input. Comparably, factor oracles (Assayag & Dubnov, 2004) are an acyclic automaton with a minimal number of states and a number of transitions linear to the length of the training sequence. They are capable of weak factor recognition, recognising all substrings in the training sequence, but also contain substrings not in the training data. Assayag and Dubnov (2004) note that factor oracles do not have a probability distribution over the alphabet at each state. However, long generated sequences should become asymptotically close to the observed training data using a generation algorithm which chooses stochastically between replicating a substring from the training corpus or jumping to a maximal suffix of the string sequence generated so far.

5.3 Corpora and Representation

To investigate methods for predicting multiple basic attributes and the effect of different smoothing techniques on a variety of data, the five symbolic datasets presented in §4.6 (Table 4-E) are used for experiments in this chapter. By using both harmonic (datasets 1 and 2) and melodic (datasets 3,4, and 5) the techniques tested in Experiments 1 (§5.4) and 2 (§5.6) can be explored more fully; particularly to discover the extent to which they are general or domain specific.

Experiments 1 (§5.4) and 2 (§5.6) use the basic harmonic and melodic viewpoints defined in §4.4.1.1 and §4.4.2 respectively. For the hamronic datasets `Root`, `ChordType`, and `PosInBar` are the basic attributes predicted, `BarLength` is not of interest for the current study. For the melodic datasets only the basic attributes `Pitch` and `Duration` are predicted. `Onset` is not predicted as it simply consists of a monotonically increasing sequence of integers.[1] The full domain of `PosInBar` in dataset 1 is $\{0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$, and $\{0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ for dataset 2.

[1] Predicting the basic attribute `Onset` with the viewpoint `Onset` is, in theory, possible with a multiple viewpoint system. However, the monotonically increasing nature of `Onset` values means that the nature of the predictions are fundamentally different to predicting the other basic attributes of the current study for a number of reasons. Firstly, there is a numerical constraint that every onset must be greater than the previous onset: $\Psi_{\texttt{Onset}}(e_1^i) > \Psi_{\texttt{Onset}}(e_1^{i-1})$. Secondly, each `Onset` value may only occur once per piece, so the STM has the unusual (but non-Markovian) constraint of not returning to a state it has seen before. See also §4.5 for further discussion.

Experiment 3 (§5.7) additionally uses viewpoints derived from `Root` and `ChordType` presented in §4.4.1.2 and §4.4.1.3 respectively.

5.4 Experiment 1: Optimal Smoothing Parameters for Harmonic and Melodic Domains

Of the statistical learning methods discussed in §5.2, PPM (used by IDyOM) offers a number of advantages over similar methods which are attractive for the current research. PPM is among the best performing lossless data compress algorithms available (Begleiter et al., 2004; Bunton, 1997; Shkarin, 2002), and whilst not the most efficient in terms of time and space complexity, this does not concern the current research which is not required to run in real time (c.f. Assayag & Dubnov, 2004; Pachet, 2003). The method does not require hand-tuned parameters (c.f. Bejerano & Yona, 2001; Ron et al., 1996) or bounded context lengths if the unbounded PPM* variant is used (Cleary & Teahan, 1997). Finally, with the PPM framework various smoothing and escape methods can be easily implemented (Bunton, 1996, ch. 6) to optimise the algorithm for different domains and corpora.

This experiment investigates the optimal smoothing parameters for linked viewpoints predicting separate attributes in a variety of harmonic and melodic datasets (Table 4-E). Pearce and Wiggins (2004) find an optimal set of smoothing parameters for a collection of monophonic melodic datasets, however, these are not guaranteed to apply to new datasets or domains. In the context of studying merged representations, it is necessary to first understand the effects of smoothing parameters on viewpoint models in different domains before comparisons are made across different forms of representation (§5.6).

5.4.1 Experimental Design

This experiment aims to find the optimal smoothing parameters for various basic attribute combinations across different datasets. Future experiments require the prediction of two attributes simultaneously, therefore, the linked viewpoint of the two basic attributes is chosen for optimisation, as opposed to the basic viewpoints individually. The harmonic datasets each have three basic attributes, giving three possible combinations of linked viewpoints to test. Model performance is assessed by mean information content, $\bar{h}$ (Equation 3.5), calculated by a 10-fold cross validation of the dataset being assessed. The mean is taken over all events, rather than over all pieces.

In theory, it is possible that each viewpoint in a multiple viewpoint system has different optimal smoothing parameters for every viewpoint predicting every target attribute. However, as the pool of possible viewpoints in a system is large,[2] the current study and previous research (Conklin & Witten, 1995; Pearce & Wiggins, 2004; Whorley, 2013) avoids this approach. Instead, all viewpoints are given the same smoothing parameters, although the LTM(+) and STM are optimised separately. The parameters to be optimised are interpolated/backoff smoothing, the escape method, and the use of update exclusion. Different global order bounds will not be investigated since unbounded length contexts (PPM*) make minimal assumptions on the dataset and domain and were found to perform consistently well by Pearce and Wiggins (2004). Furthermore, the LTM will not be used, instead, experiments will be run using the LTM+ and STM which is found to be the best combined model in Pearce and Wiggins (2004). For a given dataset and linked viewpoint, mean information content is calculated for all possible parameter combinations, with the lowest value of $\bar{h}$ signifying the optimal model.

5.4.2 Hypothesis

It is predicted that there will be some subtle differences in optimal parameters across different domains, although some techniques are expected to be universally beneficial: in particular, escape methods C, D and X. Models using interpolated smoothing are expected to outperform those that do not, and prediction models using update exclusion are predicted to be less effective than those that do not (Pearce & Wiggins, 2004).

5.4.3 Results

The results are summarised in Table 5-B, using mean information content to compare the optimal backoff and interpolated smoothing models, as well as the best models with and without update exclusion. For the LTM+ it is clear that escape methods C and D are consistently effective, however, this pattern does not extend to the STM where methods A, C, D, and X are all optimal for at least one dataset and viewpoint combination. The performance of interpolated over backoff smoothing across all datasets was assessed by taking the best performing interpolated and backoff models for each of the 18 model, dataset, and viewpoint combinations. A one-sided paired Wilcoxon signed-rank test over $\bar{h}$ values confirmed interpolated significantly outperformed backoff smoothing by 0.110 bits/symbol ($N = 18, W = 167, z = 3.550, p < 0.001$). Likewise, models that did

[2] An upper bound estimate would be $\sum_{l=1}^{L} \binom{v_n}{l}$ where v_n is the number of single viewpoints and L the maximum number of constituent viewpoints permitted in a linked viewpoint.

Table 5-B: Mean information content ($\bar{h}$) for the optimal backoff smoothing, interpolated smoothing, with update exclusion, and without update exclusion STM/LTM+ models.

Dataset	Viewpoint	Backoff smoothing		Interpolated smoothing		No Update Exclusion		Update Exclusion	
	$(\lvert[\tau_a] \times [\tau_b]\rvert)$	Model	$\bar{h}$	Model	$\bar{h}$	Model	$\bar{h}$	Model	$\bar{h}$
1	`Root⊗ChordType` (145)	STMD*U	4.372	**STMD*IU**	**4.337**	STMC*I	4.351	**STMD*IU**	**4.337**
1	`Root⊗PosInBar` (143)	STMC*	3.348	**STMC*I**	**3.312**	**STMC*I**	**3.312**	STMD*IU	3.352
1	`ChordType⊗PosInBar` (143)	STMA*U	2.657	**STMA*IU**	**2.625**	STMA*I	2.644	**STMA*IU**	**2.625**
2	`Root⊗ChordType` (121)	STMA*U	2.350	**STMD*I**	**2.263**	**STMD*I**	**2.263**	STMA*IU	2.292
2	`Root⊗PosInBar` (195)	STMA*U	2.013	**STMD*I**	**1.934**	**STMD*I**	**1.934**	STMA*IU	1.960
2	`ChordType⊗PosInBar` (165)	STMA*	1.691	**STMA*I**	**1.561**	**STMA*I**	**1.561**	STMA*IU	1.695
3	`Pitch⊗Duration` (294)	STMA*U	4.677	**STMX*I**	**4.668**	**STMX*I**	**4.668**	STMA*U	4.677
4	`Pitch⊗Duration` (629)	**STMA*U**	**4.897**	STMA*IU	4.898	STMX*I	4.950	**STMA*U**	**4.897**
5	`Pitch⊗Duration` (364)	**STMA*U**	**4.673**	STMA*IU	4.685	STMX*I	4.702	**STMA*U**	**4.673**
1	`Root⊗ChordType` (145)	LTM+C*	4.153	**LTM+C*I**	**4.080**	**LTM+C*I**	**4.080**	LTM+D*IU	4.269
1	`Root⊗PosInBar` (143)	LTM+C*	3.048	**LTM+D*I**	**2.985**	**LTM+D*I**	**2.985**	LTM+D*IU	3.157
1	`ChordType⊗PosInBar` (143)	LTM+C*	2.518	**LTM+D*I**	**2.427**	**LTM+D*I**	**2.427**	LTM+A*IU	2.714
2	`Root⊗ChordType` (121)	LTM+C*	2.694	**LTM+C*I**	**2.627**	**LTM+C*I**	**2.627**	LTM+A*IU	2.705
2	`Root⊗PosInBar` (195)	LTM+C*	2.298	**LTM+C*I**	**2.252**	**LTM+C*I**	**2.252**	LTM+A*IU	2.298
2	`ChordType⊗PosInBar` (165)	LTM+C*	1.756	**LTM+D*I**	**1.695**	**LTM+D*I**	**1.695**	LTM+A*IU	1.966
3	`Pitch⊗Duration` (294)	LTM+C*	3.628	**LTM+C*I**	**3.541**	**LTM+C*I**	**3.541**	LTM+D*IU	3.654
4	`Pitch⊗Duration` (629)	LTM+C*	4.305	**LTM+C*I**	**4.254**	**LTM+C*I**	**4.254**	LTM+D*IU	4.299
5	`Pitch⊗Duration` (364)	LTM+C*	4.350	**LTM+C*I**	**4.296**	**LTM+C*I**	**4.296**	LTM+D*IU	4.350

Note. The best model for each row is underlined and in **boldface**.

not use update-exclusion significantly outperformed those that did by 0.021 bits/symbol ($N = 18, W = 133.5, z = 2.091, p < 0.018$).

5.4.4 Conclusions and Discussion

The variety of optimal parameter combinations suggests that differing domains and viewpoints do have an impact on the effectiveness of different smoothing techniques. Therefore, when comparing models across domains and viewpoints it is necessary to optimise each first. Without optimal smoothing parameters it is difficult to attribute any results to genuine differences in statistical structure or simply the varying impact of non-optimal smoothing parameters.

In general, the relative performance of smoothing techniques established by Pearce and Wiggins (2004) was upheld. Escape methods A and B performed poorly overall, method C performed well, update exclusion was found to slightly inhibit model performance, and interpolated smoothing outperformed backoff smoothing. One notable difference is the high performance of escape method D over C when predicting `Root`⊗`PosInBar` or `ChordType`⊗`PosInBar` with the LTM+. It is also interesting to note that escape method A was optimal for four of the nine STM tests undertaken, as it has been found to perform poorly in data compression (Cleary & Witten, 1984; Moffat, 1990) and melodic prediction (Pearce & Wiggins, 2004). The instability of optimal parameters for the STMs could be attributed to the fact that the STMs themselves are highly dependant on local, dynamic statistical structure to make predictions. Therefore, the local effects of each dataset and viewpoint combination have a varying impact on the performance of different smoothing techniques. Optimal smoothing parameters for each dataset and viewpoint found in this experiment are retained for future experiments.

5.5 Merging Basic Attributes

Traditionally, multiple viewpoint systems follow Conklin (1990, p. 69) when calculating probabilities of multiple basic attribute predictions. It is assumed that the basic attributes are statistically independent, so the overall probability of multiple attributes co-occurring is simply the product of the individual probabilities. Suppose a multiple viewpoint system models two basic attributes, τ_x and τ_y, predicted by the linked viewpoint $\tau_x \otimes \tau_y$. At a given point in a sequence the system is required to predict an event represented by the tuple $\langle X, Y \rangle$ from a probability distribution over $[\tau_x] \times [\tau_y]$. Prediction is done in stages for each basic attribute to be predicted, including the matching of symbols and contexts in the PPM model. Probabilities for all symbols in $[\tau_x] \times [\tau_y]$ matching X and then matching Y are calculated. The total probability of X is the sum of all probabilities where X matches, with an identical case for the total probability of Y. Assuming statistical independence, the probability of $\langle X, Y\rangle$ is the product of the two probabilities. In other words, separate predictions are both marginalised over the other basic attribute before being multiplied:

$$p(\langle X, Y\rangle) = \sum_{y\in[\tau_y]} p(X, y) \cdot \sum_{x\in[\tau_x]} p(x, Y). \tag{5.3}$$

An alternative method is proposed which merges the basic attributes before prediction so that the probability of the merged symbol is matched and calculated directly. In this sense, X and Y are matched simultaneously and $p(\langle X, Y\rangle)$ is calculated directly by the PPM model. A merged attribute simply combines multiple basic attributes into a single representation and is modelled by a linked viewpoint. Any number of basic attributes, τ_b, may be linked to form a merged attribute which will, therefore, have a domain of $[\tau_{b_1}] \times ... \times [\tau_{b_n}]$. A merged attribute can be predicted by linked viewpoints that contain the merged attribute in their type sets. A type set is defined as the *"basic types the viewpoint is derived from and is, therefore, capable of predicting"* (Pearce, 2005, p. 59). Originally, the type set of a linked viewpoint $\tau_1 \otimes ... \otimes \tau_n$ would be $\{\tau_{b_1}, ..., \tau_{b_n}\}$ where τ_{b_1} is a basic viewpoint predicted by τ_1. A small adjustment to this definition is required to enable the prediction of merged attributes. The type set of a linked viewpoint is now the power set of its constituent viewpoints, for example, the type set of $\tau_1 \otimes \tau_2$ would be $\{\tau_{b_1}, \tau_{b_2}, \tau_{b_1} \otimes \tau_{b_2}\}$. Note that merged attributes may be predicted by linked viewpoints consisting of derived viewpoints, providing the merged attribute is contained within the type set. For example, the linked viewpoint `RootInt`⊗`MajType` may predict `Root`⊗`Chordtype`, but not `Root`⊗`PosInBar`.

Using merged attributes can be viewed as partly addressing issues concerning appropriate levels of representation. When formulating a multiple viewpoint system the appropriate basic attributes, or input representation, must be defined. In some cases it is clear that different attributes of music are clearly defined, separable dimensions, for example, pitch and duration. However, for others an appropriate representation is less clear, for example, pitch could be represented as a MIDI note (as in `cpitch`) or with two basic attributes representing pitch class and octave number. These two representations contain identical information about the musical surface, however, their statistical properties are likely to be very different.

5.6 Experiment 2: Predicting Merged Attributes from a Linked Viewpoint

In past research (Conklin & Witten, 1995; Pearce, 2005), multiple viewpoint systems have primarily been used to predict a single basic attribute (although this is not exclusively the case, e.g., Pearce et al., 2010b). The current research requires the prediction of chord symbols, comprising two attributes: `Root` and `ChordType`. This experiment empirically tests two methods for predicting multiple attributes with viewpoint systems: one that predicts attribute symbols separately and a proposed alternative that predicts merged attribute symbols. The optimal smoothing parameters for predicting merged attributes are found and then the two methods are compared.

5.6.1 Experimental Design

The experimental design broadly follows §5.4.1. For each merged attribute in the datasets the optimal smoothing parameters are found with an exhaustive search of the escape method, interpolated/backoff, and update exclusion smoothing parameters (see §5.2.1). Model performance is assessed by mean information content, $\bar{h}$, calculated with a 10-fold cross validation of each dataset. An 'M' on the end of the shorthand model description (see §5.2.1.4) signifies that the model predicts merged rather than separate attributes. By way of example, the shorthand notation for an unbounded STM using escape method AX, backoff smoothing, and update exclusion to predict a merged attribute is STMX*UM.

5.6.2 Hypothesis

As the predictive linked viewpoints are all the same as §5.4, the optimal smoothing parameters should remain similar. Interpolated smoothing is expected to outperform backoff, models without update exclusion should perform better than those with, and escape methods C and D should perform consistently well at least for the LTM+. The performance of models predicting merged attributes will be compared to predictions of separate basic attributes. When the basic attributes are statistically independent, modelling with separate basic attributes should give truer probability estimates than with merged attributes. However, when basic attributes are highly correlated, resulting in small areas of high probability density in the prediction distribution, it is expected that predicting merged attributes will outperform predicting separate attributes. This is because the merged prediction is able to take advantage of the areas of high probability density by matching both symbols directly. On the other hand, the marginalisation process required to predict separate attributes dilutes these areas of high probability in order to match both symbols independently. Therefore, it is hypothesised that when basic attributes are more correlated predicting merged attributes will be more effective.

5.6.3 Results

The optimal smoothing parameters for predicting merged attributes broadly follow the precedents established for separate attribute predictions (§5.4.3). Table 5-C shows escape methods C and D dominate the LTM+ results, while C, D, and AX are the optimal escape methods for the various STMs. Interpolated smoothing outperformed backoff smoothing by 0.058 bits/symbol ($N = 18, W = 171, z = 3.724, p < 0.001$), and non-update exclusion performed better than update exclusion models by 0.063 bits/symbol ($N = 18, W = 127, z = 1.807, p = 0.035$).

A way of quantifying correlation between basic attributes must be established in order to test the relationship between basic attribute correlation and the performance of merged attribute prediction. A chi-squared test gives a good indication of correlation between two basic attributes, however, the test statistics, χ^2, of experiments with different sample sizes cannot be compared meaningfully. Therefore, Cramer's V, $\Phi_c = \sqrt{\frac{\chi^2}{N \cdot df}}$, is used as an effect size statistic where N is the sample size (number of events), and *df* the degrees of freedom.

Additionally, a metric to quantify the difference in performance between the merged and separate attribute prediction methods is used. Paired t-tests over all pieces show that almost all differences in $\bar{h}$ are statistically significant, except for the STM of `Root`⊗`PosInBar` and `ChordType`⊗`PosInBar` for dataset 2 (Table 5-C). However, in this case t-tests are not necessarily meaningful because the sample sizes are large ($N > 150$) resulting in high t values.[3] Instead, performance difference is quantified by Cohen's d, an effect size calculated by $\frac{\bar{h_1} - \bar{h_2}}{\sigma_{pooled}}$ where σ_{pooled} is the pooled standard deviation of both populations.[4]

Figure 5.1 plots the relationship between basic attribute correlation and performance difference, confirming the general trend that more highly correlated basic attributes are better predicted as merged attributes. A linear regression confirms this trend, returning a significant effect ($df = 16, F = 11.44, p = 0.003$) and an R^2 value of 0.417.

[3] Note that it is not possible to infer from this the magnitude of the difference, only that the null hypothesis (that there is no difference between the means) can be rejected.

[4] Effect sizes can be interpreted with a loose rule of thumb, following Sawilowsky (2009): $d = 0.01$ is very small, $d = 0.2$ is small, $d = 0.5$ is medium, $d = 0.8$ is large, $d = 1.2$ is very large, and $d = 2.0$ is huge.

Table 5-C: Mean information content ($\bar{h}$) for optimal models predicting basic and merged attributes.

Dataset	Viewpoint	Separate		Merged		Correlation (Φ_c)	Difference (d)
	($\lvert[\tau_a] \times [\tau_b]\rvert$)	Model	$\bar{h}_1$	Model	$\bar{h}_2$		
1	Root⊗ChordType (145)	STMD*IU	4.337	**STMC*IUM**	**4.089**	0.325	0.176*
1	Root⊗PosInBar (143)	**STMC*I**	**3.312**	STMC*IUM	3.502	0.069	-0.235*
1	ChordType⊗PosInBar (143)	**STMA*IU**	**2.625**	STMC*IM	2.702	0.208	-0.104*
2	Root⊗ChordType (121)	STMD*I	2.263	**STMC*IM**	**1.988**	0.359	0.407*
2	Root⊗PosInBar (195)	STMD*I	1.934	**STMC*IM**	**1.928**	0.078	-0.006
2	ChordType⊗PosInBar (165)	STMA*I	1.561	**STMD*IM**	**1.529**	0.130	0.040
3	Pitch⊗Duration (294)	**STMX*I**	**4.668**	STMX*IUM	5.348	0.075	-0.904*
4	Pitch⊗Duration (629)	**STMA*U**	**4.897**	STMX*IUM	5.678	0.051	-0.736*
5	Pitch⊗Duration (364)	**STMA*U**	**4.673**	STMX*IUM	5.352	0.074	-0.822*
1	Root⊗ChordType (145)	LTM+C*I	4.080	**LTM+C*IM**	**3.679**	0.325	0.318*
1	Root⊗PosInBar (143)	LTM+D*I	2.985	**LTM+C*IM**	**2.939**	0.069	0.066*
1	ChordType⊗PosInBar (143)	LTM+D*I	2.427	**LTM+D*IM**	**2.360**	0.208	0.058*
2	Root⊗ChordType (121)	LTM+C*I	2.627	**LTM+C*IM**	**2.312**	0.359	0.409*
2	Root⊗PosInBar (195)	LTM+D*I	2.252	**LTM+C*IM**	**2.190**	0.078	0.170*
2	ChordType⊗PosInBar (165)	LTM+D*I	1.695	**LTM+D*IM**	**1.657**	0.130	0.049*
3	Pitch⊗Duration (294)	**LTM+C*I**	**3.541**	LTM+C*IM	3.647	0.075	-0.121*
4	Pitch⊗Duration (629)	**LTM+C*I**	**4.254**	LTM+C*IM	4.402	0.051	-0.152*
5	Pitch⊗Duration (364)	**LTM+C*I**	**4.296**	LTM+C*IM	4.487	0.074	-0.172*

Note. Correlation is measured by Cramer's V, $\Phi_c = \sqrt{\frac{\chi^2}{n \cdot df}}$. Performance difference is measured by Cohen's $d = \frac{\bar{h_1} - \bar{h_2}}{\sigma_{pooled}}$. Statistically significant differences (after Bonferonni correction) in performance measured with two-sided paired t-tests over all pieces at the $p < 0.001$ level are marked with *.

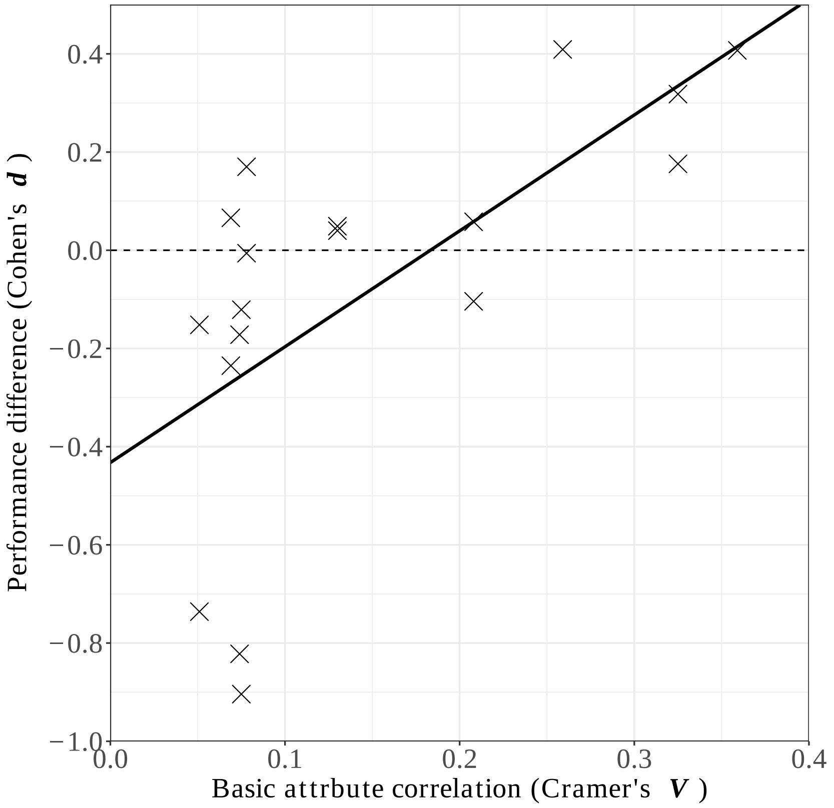


Figure 5.1: Relationship between correlation of basic attributes and relative performance of merged vs. separate attribute predictions. Merged attribute prediction outperforms separate when $d > 0$ (above the dashed line).

5.6.4 Conclusions

The main result of this experiment is that, in certain circumstances, predicting a merged attribute rather than separate attributes is more effective. Those circumstances are that the basic attributes themselves are correlated with each other, creating areas of high probability density in the distribution. A clear example of this is predicting `Root` and `ChordType` in both harmonic datasets, for both the STM and LTM+. On the other hand, when attributes are not correlated, such as `Pitch` and `Duration` in all melodic datasets, it is often more effective to predict attributes separately. This result calls into question the proposition that basic attributes can be assumed to be statistically independent from one another (Conklin, 1990, p. 69), showing they may be strongly correlated. An argument can, therefore, be made that basic attribute correlation should be measured in order to determine whether merged or separate basic attributes should be predicted by a multiple viewpoint system.

Although the overall effect of smoothing techniques on the datasets was found to hold for merged attribute prediction, there were differences in optimal parameters found between corresponding separate and merged prediction models, even though they use the same linked viewpoint model to make predictions. This may be because different smoothing techniques have differing impacts on the sparsity of models. For example, escape method A punishes the event probability more when escaping to lower orders compared to method B, so could give a sparser distribution when predicting a symbol novel to a long context. As discussed in §5.5, the relative sparsity of models is likely to have an impact on how well merged attributes are predicted.

It is interesting to note that all of the STMs for the melodic datasets found update exclusion to be effective, going against the general trend found so far in the current research. This suggests that those datasets share a property that makes counting *n*-grams with the adjusted exclusion method more effective. Since the alphabet sizes for the melodic datasets are larger than harmonic ones and an STM model is unlikely to come close to seeing all of the alphabet, the rate at which new symbols are seen is high and consequently the model often escapes down to lower order models. In this case, using an excluded count method may be beneficial as probability mass is preserved in the lower orders by excluding symbols that would have been seen in higher order models.

5.7 Experiment 3: Predicting Merged Attributes with Multiple Viewpoint Systems

Full multiple viewpoint systems such as IDyOM are complex models with several components. It is not always clear how individual components of the model will interact to give a final prediction, and therefore it follows that an improvement in one component of the model does not necessarily imply an overall improvement in performance. This final experiment tests the prediction of merged and separate attributes with a full multiple viewpoint system including viewpoint selection on the primary domain of the current research, jazz chord sequences. A full multiple viewpoint system requires a method for combining predictions from multiple models and a way of selecting a set of viewpoints which returns the lowest mean information content from the large number of possible sets (§3.5). Predictions from viewpoints and the LTM-STM models are combined with a weighted geometric mean (Equation 3.7), combining viewpoint predictions first before LTM-STM predictions as described in §3.4.4.

This experiment uses basic and derived harmonic viewpoints defined in §4.4.1 to predict the two basic attributes Root and ChordType; in other words, the chord symbol itself. PosInBar is not predicted in the current research as it is not clear whether temporal structure follows the same Markovian properties as other dimensions of music (e.g. pitch) in music perception (see §4.5). However, it is likely that temporal structure does play a role in the statistical properties, so PosInBar can be linked with other viewpoints to make predictions. As PosInBar is not being predicted its value is assumed to be known at the point of a symbol's prediction, as such it is considered to be a *given* attribute. Therefore, linked viewpoints containing PosInBar are constrained such that they match the PosInBar attribute for the predicted event.

A viewpoint pool is required which is large enough to perform well, but small enough to be computationally practical. For the current study, up to three viewpoints may be linked for a linked viewpoint, but with the condition that the linked viewpoint must include PosInBar. The primitive viewpoint pool consists of the three basic viewpoints Root, ChordType, and PosInBar, and the 10 viewpoints derived from Root and ChordType (see §4.4.1). For a system predicting Root and ChordType separately this gives a pool of 156 viewpoints and when predicting Root$\otimes$ChordType a pool of 64. Note that the difference in viewpoint pool size is due to the fact that every viewpoint in the merged attribute system must predict both attributes together, whilst for separate predictions a viewpoint need only predict one so long as the system as a whole predicts both.

5.7.1 Experimental Design

This experiment is carried out on the corpus of jazz lead sheets from the *Real Book Vol. 1* (Table 4-E dataset 1), predicting the attributes Root and ChordType. To build a full viewpoint system, viewpoint selection is undertaken first using bias weights of 7 and 2, (established by Pearce et al., 2005) for LTM-STM and viewpoint combination respectively. Afterwards, the best bias values are found for each multiple viewpoint model with an exhaustive search where $b \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 16, 32\}$. As in previous experiments, $\bar{h}$ is calculated with a 10-fold cross validation of the dataset. Throughout the experiment the optimum smoothing parameters found in Experiments 1 (§5.4) and 2 (§5.6) are retained. A model predicting separate attributes will, therefore, use STMD*IU and LTM+C*I, whilst a model predicting merged attributes will use STMC*IUM and LTM+C*IM.

The viewpoint selection algorithm is a greedy *forward stepwise selection* algorithm as described in §3.5 with minor modifications. Ordinarily, the algorithm terminates when no additions or deletions improve performance. However, Whorley (2013, pp. 189-191) notes that viewpoint selection typically results in a long tail, where the later iterations yield only small improvements in performance at the cost of time and memory complexity. Whorley (2013) curtails viewpoint selection *post hoc*; a large number of viewpoint models are processed only to be discarded. Subsequent viewpoint selections end viewpoint selection early if the difference in $\bar{h}$ between two successive iterations is less than a halting criteria threshold, typically 0.0015 bits/note (Whorley, 2013, p. 191).

For practical purposes, a method for curtailing viewpoint selection at run time is proposed for the current research (see Algorithm 1). Let $\bar{h'}_i$ be the mean information content over pieces of the current iteration and $\bar{h'}_{i+1}$ the mean information content over pieces of the proposed next iteration. Performance improvement is measured in terms of effect size by Cohen's $d = \frac{\bar{h'}_i - \bar{h'}_{i+1}}{\sigma_{pooled}}$, with the algorithm terminating if $d < 0.005$. This is the equivalent to an improvement of less than 0.5% of a standard deviation over the population of all pieces. Cohen's effect size is a parametric measure, and therefore requires h' to be normally distributed. This is not the case for the distribution of h over individual events (see Figure 5.2) which is positively skewed, but is for the distribution of h', where h' is the mean information content of a single piece. Mean information content over events ($\bar{h}$) is still used as the primary method for comparing model performance as it is not biased by the length of pieces.

5.7.2 Hypothesis

The hypothesis tested is that when using the full viewpoint system the prediction of merged attributes will outperform the prediction of separate attributes, as `Root` and `ChordType` are highly correlated in this corpus. Since the initial estimated bias weights are optimal for similar data (Pearce et al., 2005), it is likely that this difference in performance will be observed both before and after bias optimization.

5.7.3 Results

The viewpoint selection results for predicting separate and merged attributes are described in Figure 5.3. Both follow strikingly similar patterns, with five addition steps before curtailed termination on the sixth iteration. For this initial result, predicting merged attributes ($\bar{h} = 3.037$) outperformed separate attributes ($\bar{h} = 3.425$) by 0.389 bits/symbol. A paired t-test over all pieces proved this to be statistically significant ($df = 347, t = 20.587, p < 0.001$) with an effect size of $d = 0.320$. However, these results are calculated using preliminary bias parameters that have been optimised for melodic

Algorithm 1 Viewpoint selection algorithm, in the style of Cormen, Leiserson, Rivest and Stein (2001). *current-system* is an array of selected viewpoints, *viewpoints* is an array of all potential predictive viewpoints, *attributes* is an array of basic attributes to be predicted, *dataset* an array of pieces making up the training and testing data, *best-ic* is the overall mean information content of the best viewpoint system tested so far, and *best-ic-pieces* the corresponding mean information content per piece. Can-Predict returns true if the viewpoints in the *test-system* are capable of predicting all of the *attributes*. Cross-Validation runs a cross-validation of the *dataset* predicting the *attributes* with the viewpoints in *test-system* returning an array: *results*, which contain both the overall mean information content and the mean information content per piece.

```
function Select-Viewpoints(current-system, viewpoints, attributes, dataset,
                           best-ic, best-ic-pieces, threshold)
    if best-ic = NULL then
        best-ic ← $\infty$
    end if
    if best-ic-pieces = NULL then
        n ← Size(dataset)
        best-pieces-ics ← $\langle \infty_1, \infty_2 ..., \infty_n \rangle$
    end if
    for viewpoint $\in$ current-system do
        test-system ← Remove(viewpoint, current-system)
        if Can-Predict(test-system, attributes) then
            results ← Cross-Validation(test-system, attributes, dataset)
            ic ← results [0]
            ic-pieces ← results[1]
            if ic $\leq$ best-ic then
                Select-Viewpoints(test-system, viewpoints, attributes, dataset, ic,
                                  ic-pieces, threshold)
            end if
        end if
    end for
    for viewpoint $\in$ viewpoints do
        if Contains(viewpoint, current-system = FALSE) then
            test-system ← ADD(viewpoint, current-system)
            if Can-Predict(test-system, attributes) then
                results ← Cross-Validation(test-system, attributes, dataset)
                ic ← results [0]
                ic-pieces ← results[1]
                effect-size ← Cohen's-Effect-Size(ic-pieces, best-ic-pieces)
                if effect-size > threshold then
                    Select-Viewpoints(test-system, viewpoints, attributes, dataset,
                                      ic, ic-pieces, threshold)
                end if
            end if
        end if
    end for
    return current-system
end function
```

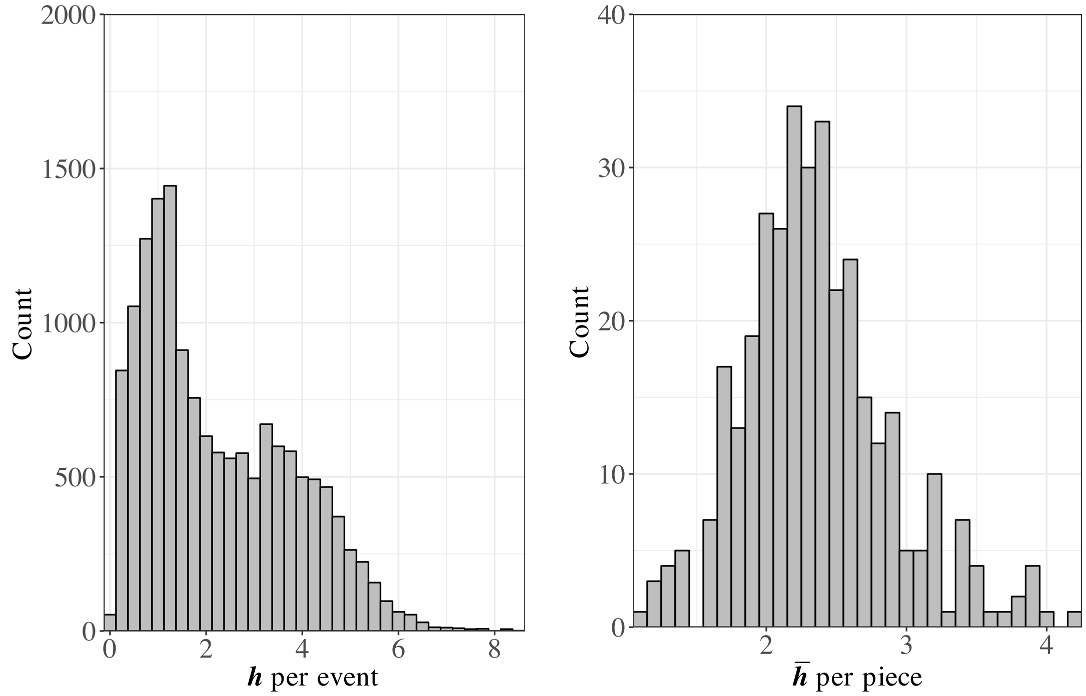
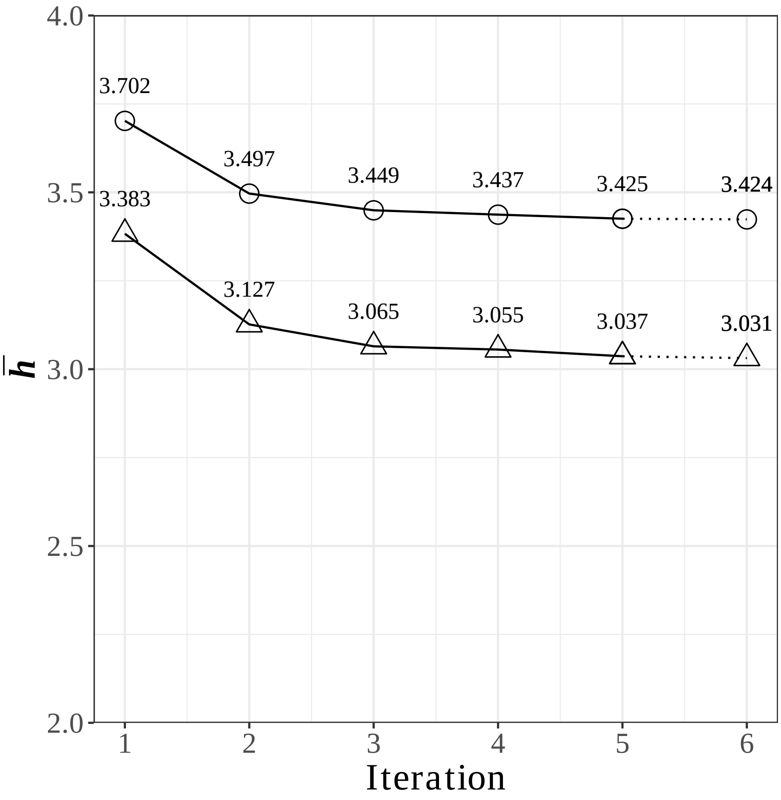


Figure 5.2: Histograms showing the distribution of information content per event (left) and per piece (right) over the *Real Book Vol. 1* database. Information content values were calculated with a STMD*IU and LTM+C*I predicting `Root` with `Root`, combining the LTM-STM distributions with a combination bias of 7.

datasets (Pearce et al., 2005). In order to make a proper comparison between the two methods, the bias parameters must be optimal for the harmonic corpus at hand.

Model performance across all bias parameters for separate prediction is shown on Table 5-D and for merged prediction on Table 5-E. A lowest $\bar{h}$ of 3.393 for separate attribute prediction was found with the LTM-STM and viewpoint biases both set to 2. The merged attribute prediction improved further with a lowest $\bar{h}$ of 2.963 with the LTM-STM and viewpoint biases set to 2 and 1 respectively, reinforcing it as the best prediction method. An improvement in performance between separate and merged attribute prediction of 0.430 bits/symbol, with an effect size of $d = 0.378$, was again found to be statistically significant ($df = 347, t = 25.529, p < 0.001$).



Viewpoint selection predicting `Root` and `ChordType` separately:
1 + `Root`⊗`ChordType`⊗`PosInBar`
2 + `RootInt`⊗`ChordType`⊗`PosInBar`
3 + `RootIntFiP`⊗`ChordType`
4 + `ChordType`⊗`PosInBar`
5 + `RootInt`⊗`RootIntFiP`⊗`PosInBar`
(6 + `Root`⊗`ChordType`)
Viewpoint selection predicting `Root`⊗`ChordType` as a merged attribute:
1 + `Root`⊗`ChordType`⊗`PosInBar`
2 + `RootInt`⊗`ChordType`
3 + `RootIntFiP`⊗`ChordType`⊗`PosInBar`
4 + `Root`⊗`ChordType`
5 + `RootInt`⊗`ChordType`⊗`PosInBar`
(6 + `RootIntFiP`⊗`ChordType`)

Figure 5.3: Viewpoint selection for multiple viewpoint systems predicting `Root` and `ChordType` as separate (circles) and merged (triangles) attributes. Viewpoints added at each iteration are shown below the graph. Parenthesised viewpoints and the dashed line indicate viewpoints added after the selection has been curtailed.

Table 5-D: Mean information content ($\bar{h}$) of a multiple viewpoint model predicting `Root` and `ChordType` separately using a range of model combination bias parameters.

		Viewpoint bias										
		0	1	2	3	4	5	6	7	8	16	32
LTM-STM bias	0	3.551	3.492	3.456	3.436	3.426	3.421	3.419	3.418	3.419	3.432	3.450
	1	3.482	3.426	**3.402**	**3.394**	**3.394**	**3.397**	**3.400**	3.404	3.408	3.428	3.446
	2	3.450	3.404	$\boxed{\mathbf{3.393}}$	**3.398**	3.409	3.420	3.429	3.437	3.444	3.474	3.493
	3	3.436	**3.398**	**3.396**	3.411	3.429	3.445	3.459	3.470	3.480	3.519	3.541
	4	3.431	**3.398**	3.403	3.424	3.446	3.467	3.484	3.497	3.509	3.556	3.581
	5	3.430	**3.402**	3.411	3.435	3.461	3.484	3.503	3.519	3.531	3.584	3.613
	6	3.432	3.406	3.418	3.445	3.473	3.498	3.518	3.535	3.549	3.607	3.638
	7	3.435	3.411	3.425	3.454	3.484	3.509	3.531	3.548	3.563	3.625	3.658
	8	3.438	3.416	3.432	3.462	3.492	3.519	3.541	3.559	3.574	3.639	3.675
	16	3.459	3.442	3.463	3.496	3.531	3.560	3.584	3.604	3.621	3.697	3.741
	32	3.476	3.461	3.485	3.521	3.558	3.589	3.613	3.634	3.652	3.734	3.780

Note. Prediction viewpoints are `Root`⊗`ChordType`⊗`PosInBar`, `RootInt`⊗`ChordType`⊗`PosInBar`, `RootIntFiP`⊗`ChordType`, `ChordType`⊗`PosInBar`, and `RootInt`⊗`RootIntFiP`⊗`PosInBar`. The lowest ten $\bar{h}$ values are in bold and the lowest boxed.

Table 5-E: Mean information content ($\bar{h}$) of a multiple viewpoint model predicting `Root⊗ChordType` using a range of model combination bias parameters.

		Viewpoint bias										
		0	1	2	3	4	5	6	7	8	16	32
LTM-STM bias	0	3.084	3.050	3.034	3.029	3.032	3.037	3.043	3.050	3.056	3.090	3.122
	1	3.000	**2.973**	**2.969**	**2.978**	2.992	3.007	3.020	3.032	3.042	3.087	3.119
	2	**2.980**	$\boxed{\mathbf{2.963}}$	**2.971**	2.993	3.018	3.042	3.062	3.080	3.094	3.153	3.188
	3	**2.980**	**2.970**	**2.986**	3.015	3.047	3.077	3.102	3.124	3.142	3.213	3.252
	4	2.988	**2.981**	3.001	3.035	3.071	3.105	3.134	3.158	3.178	3.259	3.303
	5	2.996	2.992	3.015	3.052	3.091	3.126	3.157	3.184	3.206	3.294	3.341
	6	3.005	3.002	3.027	3.065	3.106	3.144	3.176	3.204	3.227	3.321	3.371
	7	3.013	3.011	3.037	3.077	3.119	3.157	3.191	3.219	3.244	3.342	3.394
	8	3.020	3.019	3.045	3.086	3.129	3.168	3.203	3.232	3.257	3.359	3.413
	16	3.054	3.054	3.082	3.127	3.173	3.215	3.252	3.284	3.313	3.429	3.491
	32	3.078	3.078	3.107	3.154	3.201	3.245	3.283	3.316	3.347	3.471	3.537

Note. Prediction viewpoints are `Root⊗ChordType⊗PosInBar`, `RootInt⊗ChordType`, `RootIntFiP⊗ChordType⊗PosInBar`, `Root⊗ChordType`, and `RootInt⊗ChordType⊗PosInBar` using a range of model combination bias parameters. The lowest ten $\bar{h}$ values are in bold and the lowest boxed.

5.7.4 Conclusions

The main result of this experiment is that the prediction of merged attributes outperforms predicting the same attributes separately in a dataset where those attributes are highly correlated. The difference in performance of 0.430 bits/symbol is substantial, equivalent to 37.8% ($d = 0.378$) of the pooled standard deviations of the population of pieces and warrants consideration in future research using multiple viewpoint models. Interestingly, this is greater than both of the improvements in performance for the STM and LTM+ found in Experiment 2. It seems that in this instance the improvements over separate prediction seen in the individual models are exaggerated by the extra components of the full multiple viewpoint system, rather than diminished.

Both methods resulted in similar multiple viewpoint models through viewpoint selection, presumably because the PPM* components of both merged and separate methods are identical. `Root`, `RootInt`, and `RootIntFiP` were present in both viewpoint selections and in both cases were selected in that order, giving a strong indication of importance (since the algorithm greedily selects viewpoints). In contrast to viewpoint selection of melodic data (Pearce, 2005, pp. 127-128), the root interval viewpoints (`RootInt, RootIntFiP`) are poorer predictors than `Root` itself. Pearce (2005) found that `cpint` was an important predictor of `cpitch` as it generalised statistical structure by allowing transpositionally equivalent sequences to be considered the same. The fact that this does not appear to translate to the harmonic domain could be attributed to a mixture of three factors. Firstly, the alphabet size of `Root` is 13, considerably smaller than the typical alphabet size of `cpitch` (21 to 37 for the datasets in the current study). Large alphabets have a problem of sparsity, partially solved by generalising with interval viewpoints, however, this need not be a problem for the small alphabet of `Root` symbols. Secondly, given that `Root` models are not particularly sparse, it may be the case that most of the common harmonic progressions occur in most (or all) transpositions in the dataset. As this happens, there will be enough statistical structure in the model for any given transposition without having to revert to a derived viewpoint such as `RootInt` to describe it successfully. Finally, the occurrence of the special *NC* root symbol slightly damages the predictive power of the `RootInt` viewpoint. Since it does not have a proper pitch class value, `RootInt` must divide the prediction probability over the whole alphabet (Pearce, 2005, p. 115). This final effect is unlikely to be particularly large: for dataset 1 only 0.863% of chords (131 of 15,197) are *NC*.

None of the viewpoints derived from `ChordType`, namely `MajType`, `7Type`, and `FunctionType`, were selected for either model. These viewpoints simplified sequences by categorising `ChordType` into smaller categories. Viewpoints such as these are only successful if the information gained by generalising sequences of sparse data outweighs the information lost when converting from the derived sequences back to the event space, ξ. It appears this was not the case for the current study, suggesting that `ChordType` is not particularly sparsely[5] distributed in the jazz dataset.

The appearance of `PosInBar` twice in each model suggests that temporal structure is correlated to a degree with harmonic structure for the corpus. This implies that certain harmonic functions are more likely to occur on different beats of the bar, for example, a tonic may be likely to occur on the first beat of a bar.

The bias combination results contrast strikingly with those established for melodic corpora. Notably, Pearce et al. (2005) establishes a high LTM-STM bias of 7 to predict `cpitch`, whilst the current study finds an LTM-STM bias of 2 predicts `Root` and `ChordType` best (both merged and separately). The suggestion that the high LTM-STM bias helps to weight away from cases where the STM produces high entropy predictions because of a lack of context (see Pearce, 2005) does not appear to apply to the current domain and dataset. It is possible that the STM in general performs well for the jazz dataset as there is often a large amount of repetition within a lead sheet.

5.8 Summary and Discussion

This chapter has presented and tested contrasting methods for the prediction of multiple basic attributes with multiple viewpoint systems across three experiments. One method predicts basic attributes separately, whilst another forms a merged representation so that simultaneous predictions can be made. As hypothesised, it was found that when the basic attributes are highly correlated in a corpus they are better predicted by the merged method, whereas if they are relatively uncorrelated a separate model is best (§5.6). With a full multiple viewpoint system predicting the primary domain of the study, jazz chord sequences, the merged method statistically significantly outperformed predicting separate attributes by 0.430 bits/symbol (§5.7). These results strongly imply that basic attributes should not be considered as statistically independent, in contrast to assumptions in the early multiple viewpoint literature (Conklin, 1990, p. 69). Rather, multiple viewpoint systems may take advantage of the statistical regularities arising from potential correlations between certain basic attributes. Furthermore, calculating the probability of multiple surface attributes as the product of the individual basic attributes is not guaranteed to be an accurate estimate of probability.

A secondary contribution of this chapter was to explore the optimal smoothing tech-

[5] Figure 4.4 shows `ChordType` to have a distribution with no zero values, although with a low entropy.

niques for melodic and harmonic domains predicted by separate and merged attributes. Experiment 1 (§5.4) reinforced that techniques found to be effective for monophonic melodic prediction (Pearce & Wiggins, 2004) performed well when predicting chord sequences. Interpolated smoothing statistically significantly outperformed backoff smoothing. The use of update exclusion was in general found to damage model performance. One notable difference was that escape method D performed well for certain harmonic attribute combinations involving temporal information (`PosInBar`). Escape method C predicted the melodic datasets and chord symbols (consisting of `Root` and `ChordType`) effectively, confirming the results established by Pearce and Wiggins (2004). These results held for the prediction of merged attributes in Experiment 2 (§5.7) with only minor discrepancies. A useful avenue for future research might be to investigate whether the optimal smoothing techniques for a dataset can be ascertained from the dataset's features and qualities. The chief predictors might be the alphabet size, a measure of the zero-order distribution of the dataset (Shannon entropy), and the rate at which new symbols are seen. The problem could be approached as a supervised machine learning task, with those predictors as variables, and the optimal set of smoothing parameters found with exhaustive search.

The effectiveness of predicting merged attributes for highly correlated basic attributes hints at some interesting implications for cognitive representation. In hand constructed multiple viewpoint systems (Conklin & Witten, 1995), as well as for viewpoint selection (Pearce, 2005, pp. 127-128), it has been speculated that linked viewpoints are effective predictors when their constituent viewpoints are correlated. The current research supports this idea but goes further, suggesting that correlated attributes are merged not only at the prediction level (the linked viewpoint), but also on the surface level. How this translates onto a cognitive system is not clear, as it would be naive to directly map processes within IDyOM onto human cognition. Tentatively, it could be hypothesised that representations that are found to be correlated are merged into a single representation. Certainly, from a computational perspective this study has shown that this gives a more compact representation in terms of information theoretic properties; a lower mean information content implies closer fits between the model and training data. A merged representation contains identical absolute information about the surface compared to separate representations; no information is lost since a merged representation is simply a Cartesian product of its constituent attributes. However, the representation is more compact owing to a lower mean information content, therefore, it is estimated that less bits are required to represent each event. This gain in representational efficiency does not increase time or space complexity since the predictive part of the model (the viewpoints) are identical for both methods, only the surface representation is different.

Chapter 6

Improving Predictions of Derived Viewpoints

6.1 Overview

This chapter deals specifically with the predictive power of derived viewpoints. A potential problem with derived viewpoint prediction is identified in §6.2, whereby derived viewpoints may lose information during the mapping between the derived and basic domains. Ordinarily, when a single derived element maps onto multiple basic elements the probability mass is divided uniformly between the basic elements. §6.4.3 proposes a solution to this problem by weighting the probabilities associated with this mapping by their zero-order counts. This method is tested on individual (§6.4.1) and multiple (§6.4.2) viewpoint systems. In addition, the impact of weighting the inverse function on model compactness is assessed in §6.4.3.

6.2 Problems Associated with Derived Viewpoint Prediction

Derived viewpoints[1] aim to aid predictions of multiple viewpoint systems by using musical structure to abstract away from the musical surface. This abstraction typically takes one of two forms. Firstly, a derived viewpoint may categorise basic elements with a surjective function providing a many-to-one mapping between individual elements in

[1] See §3.3.2 for a technical review, and §4.4.1.2, §4.4.1.3, and §4.4.1.4 for the derived viewpoints used in the current research.

the basic domain onto the derived domain. Examples of such viewpoints are `MajType`, `7Type`, and `FunctionType`, which apply different categorisations for `ChordType` in the current research (§4.4.1.3), or `cpitch-class` in the research of Pearce (2005), which groups pitches into pitch class categorises, effectively applying octave equivalence. The second form of derived viewpoint finds relational structure between events in a sequence, usually the interval between two pitches in melodic research, or between two roots in harmonic research. The relation may be between adjacent events, such as in `RootInt` (§4.4.1.2) and `cpint` (Pearce, 2005), or non-adjacent as in `RootIntFiP` (§4.4.1.2) and `cpintfip` (Pearce, 2005). For these viewpoints the form of the abstraction is effectively reducing the length of the sequence that contains useful information. For example, a sequence of n `Root` elements is reduced to a sequences of $n-1$ `RootInt` elements since the first event is undefined ($\perp$). Similarly, when `RootIntFiP` is applied to a sequence of n `Root` elements, the first derived element of the sequence will always be 0 (the interval between the first root and itself). In effect, this reduces the useful information in the sequence to $n-1$ elements. This chapter identifies and proposes solutions to problems associated with the first form of derived viewpoint, those which categorise basic elements. Chapter 8 studies the second, relational, form of derived viewpoint in more depth.

Derived viewpoints that capture useful structure in these abstractions are able to generalise training data, reducing sparsity and forming better predictions of the test data. Inevitably, some information is lost in this generalisation of data. Specifically, since probabilistic models (§3.4.1) are built of sequences in the derived domain $[\tau]^*$, and predictions made over the basic event space ξ^*, some information will be lost in the mapping from the derived to basic domains. This mapping is implemented by the inverse viewpoint function Ψ' (see §3.4.5). When the loss of information from the inverse viewpoint function is outweighed by the gain in generalisation of data, the derived viewpoint will outperform the basic viewpoint it is derived from as measured by mean information content, $\bar{h}$.

With respect to the present research, viewpoint selection results over the *Real Book Vol. 1* (dataset 1, Table 4-E) show viewpoints derived from `ChordType` are not selected (§5.7) when predicting `Root`$\otimes$`ChordType`. This suggests that potentially two factors are in play: that the viewpoints derived from `ChordType` do not generalise information well, and that they lose too much information when mapping back to the basic domain.

6.3 Using Zero-order Statistics to Weight Ψ'

Firstly, it is useful to show in detail cases where certain derived viewpoints would be poor predictors for a basic attribute. Where a derived viewpoint maps an element onto a large number of basic elements, a certain amount of information is lost by dividing the probability mass uniformly. Suppose a prediction from `MajType` returns a high probability for a major chord, mapping onto a *'7'*, *'M7'*, *'6'*, *'alt'*, or *'aug'* `ChordType`. *'7'* and *'M7'* chords are very common, whilst *'alt'* and *'aug'* chords are comparatively rare. Since `MajType` must distribute probability mass equally to all five of these basic elements, a considerable amount of information is lost and it remains a poor predictor of `ChordType`. The predictive strength of these kinds of viewpoints is to generalise data that will become sparse, specifically in sequence prediction when matching contexts in the PPM* model. This strength is likely to be reduced by the uniform allocation of probability mass and could make these viewpoints poor predictors; returning high mean information content estimates and causing the viewpoints to remain unselected in viewpoint selection.

A general approach to counter this loss of information is to weight probabilities with the zero-order (unigram) frequencies when distributing probability mass from a derived element to the relevant basic elements. For reference, Equation 6.1 shows a probability estimate of a basic element, $p(t_{\tau_b})$, calculated by uniformly distributing the probability mass of a derived element, $p(t_\tau)$, following Pearce et al. (2005). B represents the set of basic elements that are mapped onto from the derived element t_τ. The proposed alternative, shown in Equation 6.2, uses probabilities from the zero-order model $p_0(t_{\tau_b})$ to weight the distribution of probability mass from t_τ to t_{τ_b}. As with PPM* predictions, probability mass must be reserved for unseen symbols in the basic element alphabet, so a smoothing method and -1^{th} order distribution is utilised. Using an established smoothing framework (Pearce & Wiggins, 2004), Equation 6.3 shows an interpolated smoothing method (see Equation 5.2) with escape method C (see Table 5-A), an order bound of 0 and with no update exclusion. $c(t_{\tau_b})$ is the number of times the symbol t_{τ_b} occurs in the training set, J is the length of the training set, $[\tau_b]$ is the alphabet of the basic viewpoint, and $[\tau_b]'$ the observed alphabet of the basic viewpoint.

$$p(t_{\tau_b}) = \frac{p(t_\tau)}{|B|} \tag{6.1}$$

$$p_w(t_{\tau_b}) = p(t_\tau)\frac{p_0(t_{\tau_b})}{\sum_{u \in B} p_0(u)} \tag{6.2}$$

$$p_0(t_{\tau_b}) = \frac{c(t_{\tau_b})}{J + |[\tau_b]'|} + \frac{|[\tau_b]'|}{J + |[\tau_b]'|} \cdot \frac{1}{|[\tau_b]| + 1 - |[\tau_b]'|} \tag{6.3}$$

A demonstration of this process is shown in Figure 6.1. `FunctionType` is used to predict the next `ChordType` symbol with an LTM model given the context *Am7, D7, Bm7, Bbm7*. The top chart shows a strong expectation of a pre-dominant chord, which could map onto a *m7*, *halfdim*, or *dim* `ChordType`. With an unweighted Ψ' (Equation 6.1) mapping from `FunctionType` to `ChordType`, these three basic elements are all given equal probability (middle chart). However, since *m7* is far more common than *halfdim* and *dim*, a more accurate probability distribution could be one weighted (Equation 6.2) by the zero-order frequencies (bottom chart), assigning a high probability to *m7*. This approach allows the powerful generalisation of derived viewpoint models to be combined efficiently with more specific predictions from the basic viewpoint.

6.4 Testing the Impact of Weighting Ψ'

To investigate the effect of weighting Ψ'_τ with a zero order model, the mean information content, $\bar{h}$ (Equation 3.5), is used as a performance metric to compare predictions with the weighted and unweighted inverse mapping function. As always, $\bar{h}$ is calculated with a 10-fold cross-validation of the corpus. The effect of the weighting on individual derived viewpoints is observed first (§6.4.1) before comparing the impact on full multiple viewpoint systems (§6.4.2).

For the individual viewpoints, it is expected that derived viewpoints that categorise, and abstract heavily from their basic viewpoint will benefit most from weighting Ψ'. Typically, these are viewpoints derived from `ChordType`, for example, `MajType` reduces the alphabet of `ChordType` from 13 down to 3. By contrast, it is expected that the impact of weighting Ψ' will be far smaller, if significant at all, for derived viewpoints which are relational, and have a close to one-to-one mapping between alphabets (e.g. `RootInt`). When constructing a full multiple viewpoint system it is hoped that weighting Ψ' will help for more derived viewpoints to be selected during viewpoint selection. Not only should this give a lower mean information content, but also produce a more compact viewpoint model. Successful derived viewpoints should abstract information away from basic viewpoints onto smaller alphabets without a loss in performance.

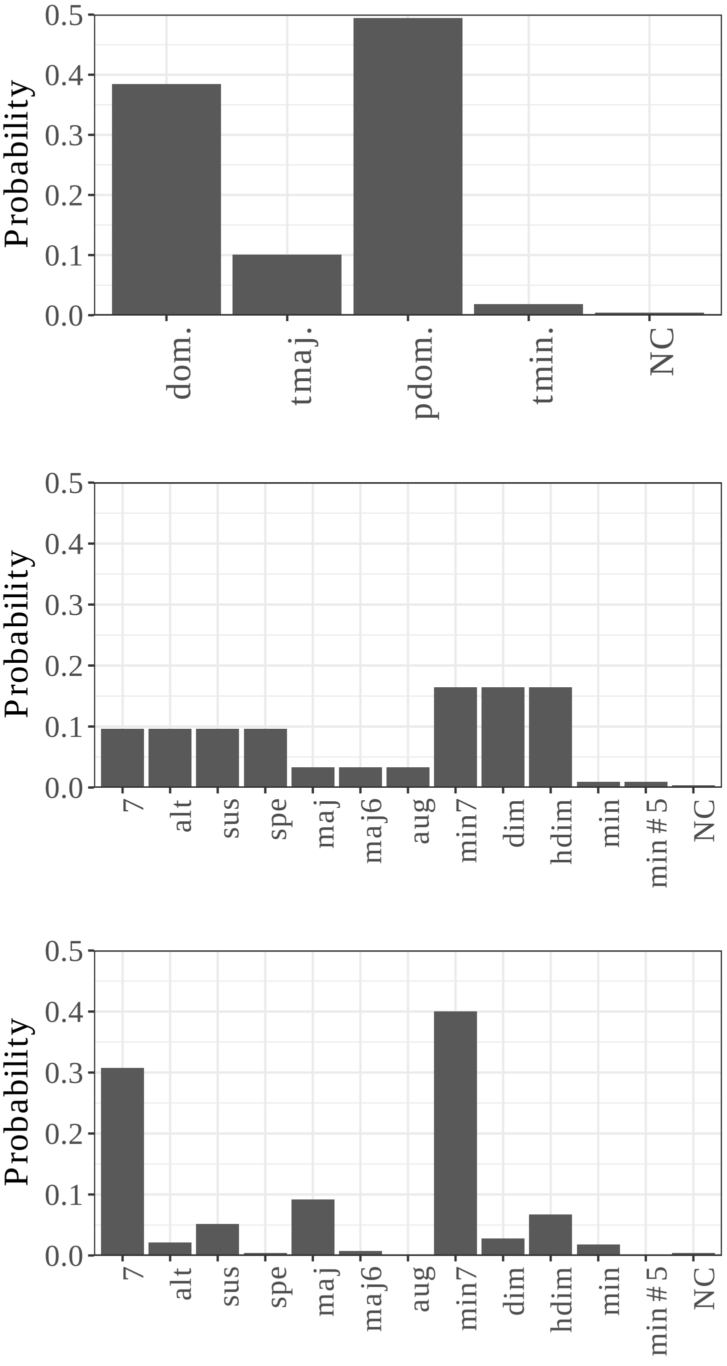


Figure 6.1: Top: probability distribution of `FunctionType` following the context *Am7, D7, Bm7, Bbm7*. Middle and bottom: probability distributions for `ChordType` predicted by `FunctionType` with an unweighted (middle) and zero-order weighted Ψ' (bottom).

6.4.1 Individual Viewpoint Results

Six derived viewpoints for predicting `Root` and `ChordType` are chosen for testing, as well as the basic viewpoints themselves for reference. The smoothing parameter configuration is STMD*IU-LTM+C*I (see §5.2.1.4), and LTM-STM predictions are combined with a weighted geometric mean as described in §3.4.4. A bias weight of $b = 2$ for the LTM-STM combination is used. All of these free parameter selections are based on the findings established in §5.4 and §5.7 when predicting separate viewpoints. The *Real Book Vol. 1* (dataset 1, Table 4-E) is the corpus under investigation.

Table 6-A shows the mean information content calculated using both weighted and unweighted Ψ' functions. Effect size measured by Cohen's $d = \frac{\bar{h_1}-\bar{h_2}}{\sigma_{pooled}}$ across all pieces ($n = 348$) is used to quantify the relative performance for each viewpoint. A one-sided paired t-test across pieces assesses statistical significance between the means at the $p < 0.001$ level, marked with a *. Bonferroni correction is used to account for the 8 repeated statistical tests, so the corrected significance level is $p < 1.25 \times 10^4$.

Strikingly, the derived viewpoints predicting `ChordType` benefit most from the weighting method, all with effect sizes greater than 1.0 and an absolute improvement of over 1.0 bit/symbol. By contrast, the impact of weighting Ψ' on the viewpoints derived from `Root` is small and inconsistent, with effect sizes of around 0.1 or less, and the improvement for `RootInt` found to be insignificant at the $p < 0.01$ level ($df = 347, t = 2.294, p = 0.011$). It is likely that this is because in the majority of cases `RootInt` has a one-to-one mapping with `Root`, except for the *NC* case where a `RootInt` symbol of -1 maps onto the full alphabet of `Root`. It is interesting to note that none of the individual derived viewpoints are able to predict their basic viewpoint better than the basic viewpoint itself, even with a weighted Ψ'. This suggests that they may not be selected in the viewpoint selection, which will greedily select viewpoints that produce the lowest $\bar{h}$. However, linked viewpoints have not yet been tested, and therefore at this point their impact on full multiple viewpoint systems is unknown, and must be tested with the viewpoint selection algorithm.

6.4.2 Viewpoint Selection Results

The viewpoint selection algorithm (Algorithm 1) is employed to investigate the impact of weighting Ψ' on full multiple viewpoint systems. Briefly, viewpoints are greedily selected with a forwards stepwise algorithm with $\bar{h}$ as a heuristic. Selection is curtailed if the improvement in $\bar{h}$ has a Cohen's effect size of $d < 0.005$, measured as a distribution over pieces rather than events. The task is aligned to the primary goal of the present

Table 6-A: Predicting ChordType (top) and Root (bottom) with weighted and unweighted Ψ'.

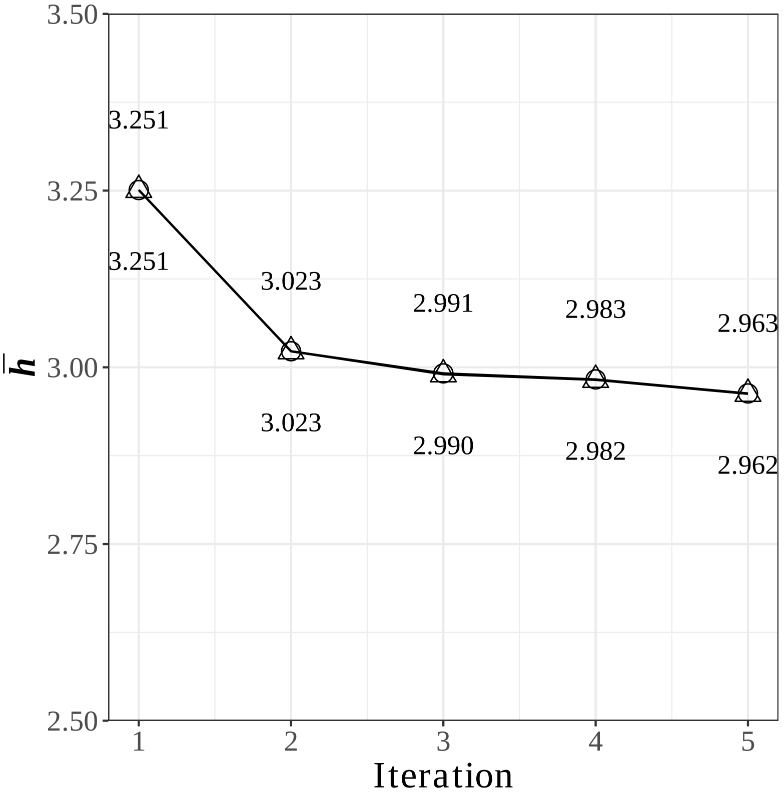
Derived Viewpoint	Unweighted Ψ'	Weighted Ψ'	d
ChordType	1.792	1.792	0.000
MajType	3.268	2.141	2.396*
7Type	3.248	2.171	2.235*
FunctionType	3.061	1.972	2.185*
Root	2.251	2.251	0.000
RootInt	2.290	2.276	0.030
MeeusInt	3.141	2.934	0.370*
ChromaDist	2.690	2.629	0.117*

Note. Performance difference is measured by Cohen's $d = \frac{\bar{h}_1 - \bar{h}_2}{\sigma_{pooled}}$. * marks differences that are statistically significant (after Bonferonni correction) at the p <0.001 level according to a one-sided paired t-test.

research, predicting harmonic sequences in jazz music. Root⊗ChordType is the merged attribute (§5.5) to be predicted over the *Real Book Vol. 1* dataset (dataset 1, Table 4-E), whilst PosInBar is a given attribute (see §4.5). Following the optimal results from §5.6 and §5.7 a STMC*IUM-LTM+C*IM model uses bias weights of $b = 2$ and $b = 1$ for LTM-STM and viewpoint combination respectively, which are both carried out with a weighted geometric mean.

The viewpoint selection algorithm is run twice; once each for the weighted and unweighted Ψ' models. Figure 6.2 shows that both weighted and unweighted Ψ' models select identical viewpoints at each iteration. The difference in performance of $\bar{h} = 2.963$ for unweighted Ψ' and $\bar{h} = 2.962$ for weighted Ψ' is negligible, with a Cohen's effect size of $d = 0.001$, confirmed as not significant at the $p < 0.001$ level with a t-test over pieces ($df = 347, t = 1.297, p = 0.098$).

The weighting of Ψ' does not enable any of the viewpoints derived from ChordType that performed successfully in §6.4.1 to be selected. Therefore, it can be assumed that the linked viewpoints perform similarly to their related primitive derived viewpoints, i.e. weighting Ψ' considerably improves their predictive power but not to the extent that they outperform the basic viewpoint ChordType. With this is mind, it is unsurprising that the performance difference between the models is not significant, since the only derived viewpoints are those derived from Root, and do not particularly benefit from weighting Ψ'.



Viewpoints selected with both an unweighted and weighted Ψ'.
1 + `Root`⊗`ChordType`⊗`PosInBar`
2 + `RootInt`⊗`ChordType`
3 + `RootIntFiP`⊗`ChordType`⊗`PosInBar`
4 + `Root`⊗`ChordType`
5 + `RootInt`⊗`ChordType`⊗`PosInBar`

Figure 6.2: Viewpoint selection for multiple viewpoint systems predicting `Root`⊗`ChordType` with unweighted (circles, labels above) and weighted (triangles, labels below) Ψ'. Viewpoints added at each iteration are shown below the graph (note that both viewpoint selection runs select identical viewpoint systems).

6.4.3 Using Ψ' for Compact Multiple Viewpoint Systems

The compactness of multiple viewpoint systems is relevant both to computational complexity and their relationship with cognitive representations. Searching a suffix tree with the PPM* algorithm with the current implementation using Ukkonen's algorithm (Ukkonen, 1995) is achieved in linear time (linear to the size of the training data J), but must be done $|[\tau]|$ times to return a complete prediction set over the viewpoint alphabet $[\tau]$, giving a time complexity of $O(J|[\tau]|)$ (see §3.6 for a more in-depth discussion on the computational complexity of multiple viewpoint systems). Selecting viewpoints with a smaller alphabet size has, therefore, a substantial impact on the time complexity for the system. From the perspective of computational models for human cognition and perception (Pearce & Wiggins, 2012), selecting viewpoints with smaller alphabets with only a small, acceptable loss of performance is similar to building levels of abstraction when learning cognitive representations (Wiggins & Forth, 2015). The question of 'how much is an acceptable loss of predictive performance?' is beyond the scope of the current research. However, it can be noted that humans appear to accept a slight loss of information, but not too much, when building cognitive representations that are more compact. A prime example is the development of relative pitch representations in adults, compared to infants who primarily rely on absolute pitch (Saffran & Griepentrog, 2001; Saffran et al., 1999). However, contour as a pitch representation is highly compact, although not fine-grained enough to distinguish between most melodies (Trehub, Bull & Thorpe, 1984). This suggests that, whilst contour is a highly compact representation for pitch consisting only of three symbols, it is insufficient for carrying out straightforward perceptual tasks. Absolute pitch, on the other hand, is highly descriptive, but at the cost of a large alphabet. Relative pitch, meanwhile, works on an intermediate level of abstraction that appears to be cognitively advantageous.

A brief computational analysis is presented to quantify the loss in performance of a computational model using representations of differing compactness. Performance is measured by $\bar{h}$, and the computational models in question are multiple viewpoint models predicting `Root`$\otimes$`ChordType` over the *Real Book Vol. 1* (dataset 1, Table 4-E) with the same model configurations as §6.4.2. Model compactness, $C(M)$, is measured in terms of the total number of viewpoint elements in the predictive viewpoint domains (Equation 6.4) of a multiple viewpoint system M, comprised of K viewpoints.

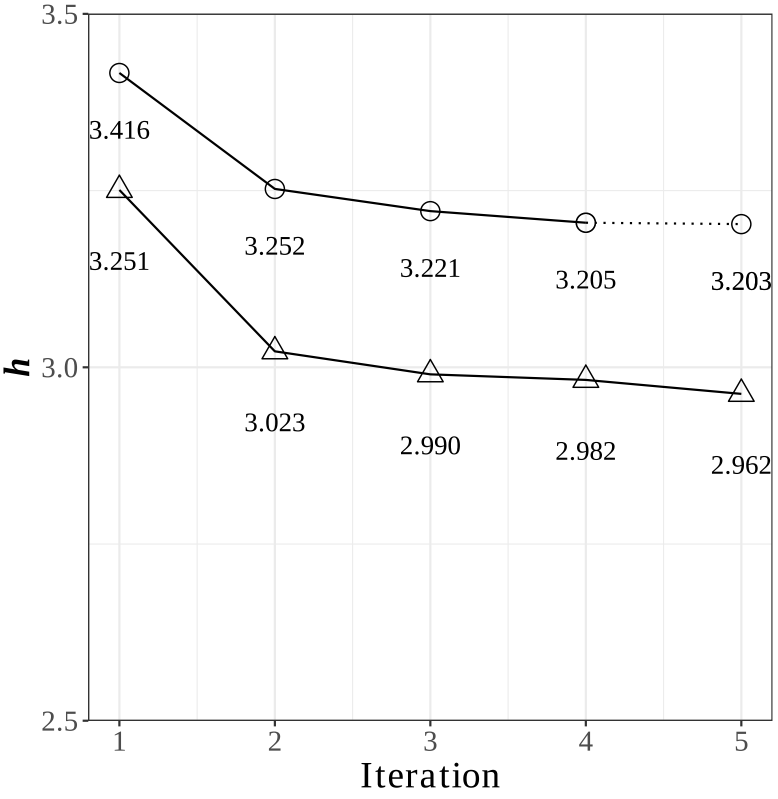
$$C(M) = \sum_{i=1}^{K} |[\tau_i]| \tag{6.4}$$

A restricted viewpoint selection is introduced to force models to use viewpoints derived from `ChordType`, rather than `ChordType` itself, in theory producing more compact multiple viewpoint systems. `ChordType` is chosen over `Root` as the basic viewpoint because its derived viewpoints benefited most from weighting Ψ' (§6.4.1). A restricted viewpoint selection simply removes all viewpoints containing `ChordType` from the available viewpoint pool. When predicting `Root`$\otimes$`ChordType` this reduces the pool from 64 to

48 viewpoints. The model parameters and task match §6.4.2, a STMC*IUM-LTM+C*IM model uses bias weights of $b = 2$ and $b = 1$ for LTM-STM and viewpoint combination respectively (carried out with a weighted geometric mean). The purpose of the analysis is simply to compare a restricted and unrestricted viewpoint selection run, both with weighted Ψ', and compare the differences in model performance ($\bar{h}$) and compactness, $C(M)$ (Equation 6.4).

Both viewpoint selections are summarised in Figure 6.3. The unrestricted selection (an identical run to §6.4.2) returns a performance of $\bar{h} = 2.962$ bits/symbol. This outperforms the restricted selection ($\bar{h} = 3.205$ bits/symbol) significantly ($df = 347, t = 24.305, p < 0.001$) with a two-sided t-test over pieces. The absolute difference is relatively small (0.243 bits/symbol), as is the Cohen's effect size ($d = 0.222$). By contrast, the restricted viewpoint selection produces a model that is far more compact ($C(M) = 234$) compared to the unrestricted selection ($C(M) = 845$), which is over 3.6 times greater. The result is a multiple viewpoint system that is far more compact, at the expense of a small but noticeable drop in performance.

It is hypothesised that such a system requires Ψ' to be weighted in order for viewpoints derived from `ChordType` to be effective. To test this, the restricted and unrestricted viewpoint selections are repeated with an unweighted Ψ' (Figure 6.4). When Ψ' is unweighted the difference in performance is extremely prominent: $\bar{h} = 2.963$ for the unrestricted selection, and $\bar{h} = 4.309$ for the restricted viewpoint selection. The difference (1.346 bits/symbol) is statistically significant as measured by a two-sided t-test over pieces ($df = 347, t = 71.253, p < 0.001$) with a very large Cohen's effect size of $d = 1.330$. In terms of model compactness, the restricted version is again more compact, returning $C(M) = 337$ in comparison to the unrestricted version where $C(M) = 845$. This ratio of 2.5 for the unweighted Ψ' models is smaller than the ratio of 3.6 for the weighted Ψ' versions. However, since the stopping criteria (when the effect size is $d < 0.005$) for viewpoint selection is somewhat arbitrary, and the restricted viewpoint selection with unweighted Ψ' continues to add viewpoints which lower $\bar{h}$ by only a small amount, a fairer comparison between the models can be made by the first five viewpoints selected only. In this case the difference in performance is slightly greater (1.360 bits/symbol), is statistically significant ($df = 347, t = 71.438, p < 0.001$), and shows a very large effect size of $d = 1.345$. Difference in compactness increases: the restricted version ($C(M) = 273$) is now 3.1 times more compact than the unrestricted version ($C(M) = 845$). Therefore, the unweighted Ψ' versions produces models which gain less in compactness, and lose more in performance compared to their weighted Ψ' counterparts. To conclude, a weighted Ψ' is necessary to avoid large drops in performance when constructing compact multiple viewpoint systems, utilising specific sets of derived viewpoints.



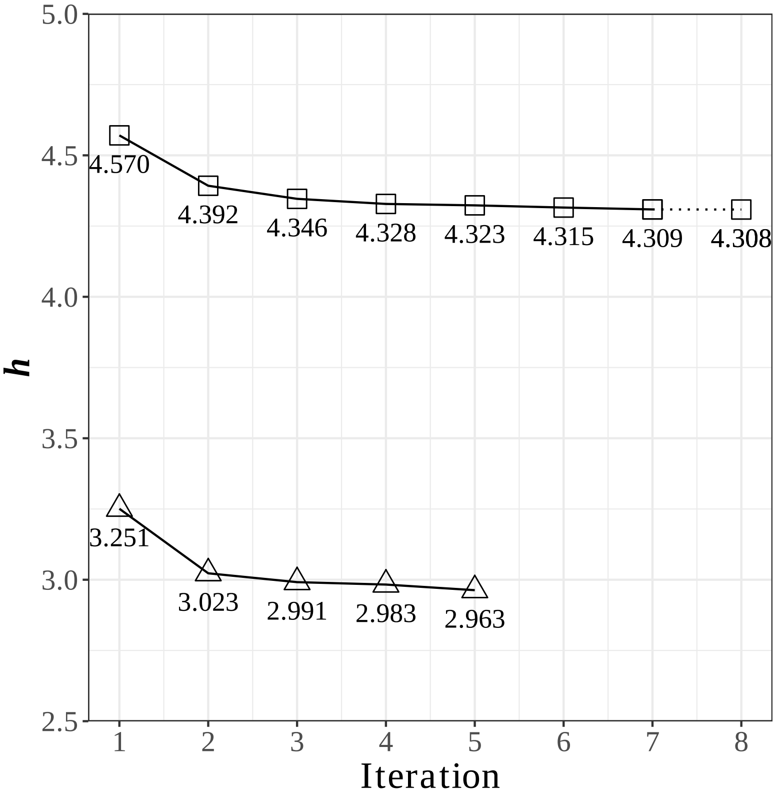
Viewpoints selected with a weighted Ψ′ and an unrestricted viewpoint pool (triangles).

1 + `Root`⊗`ChordType`⊗`PosInBar`
2 + `RootInt`⊗`ChordType`
3 + `RootIntFiP`⊗`ChordType`⊗`PosInBar`
4 + `Root`⊗`ChordType`
5 + `RootInt`⊗`ChordType`⊗`PosInBar`

Viewpoints selected with a weighted Ψ′ and a restricted viewpoint pool excluding `ChordType` (circles).

1 + `Root`⊗`FunctionType`⊗`PosInBar`
2 + `RootInt`⊗`FunctionType`⊗`PosInBar`
3 + `RootIntFiP`⊗`FunctionType`
4 + `RootInt`⊗`MajType`⊗`PosInBar`
(5 + `Root`⊗`FunctionType`)

Figure 6.3: Viewpoint selections for multiple viewpoint systems predicting `Root`⊗`ChordType`. Viewpoint selection is run on an unrestricted viewpoint pool (triangles), and a restricted viewpoint pool excluding `ChordType` (circles), both with a weighted Ψ′. Viewpoints added at each iteration are shown below the graph, viewpoints added after selection has been curtailed are parenthesised and plotted with a dashed line.



Viewpoints selected with an unweighted Ψ' and an unrestricted viewpoint pool (triangles).

1 + `Root`⊗`ChordType`⊗`PosInBar`
2 + `RootInt`⊗`ChordType`
3 + `RootIntFiP`⊗`ChordType`⊗`PosInBar`
4 + `Root`⊗`ChordType`
5 + `RootInt`⊗`ChordType`⊗`PosInBar`

Viewpoints selected with an unweighted Ψ' and a restricted viewpoint pool excluding `ChordType` (squares).

1 + `Root`⊗`FunctionType`⊗`PosInBar`
2 + `RootInt`⊗`MajType`⊗`PosInBar`
3 + `RootIntFiP`⊗`FunctionType`
4 + `RootInt`⊗`7Type`⊗`PosInBar`
5 + `Root`⊗`FunctionType`
6 + `RootInt`⊗`FunctionType`⊗`PosInBar`
7 + `Root`⊗`MajType`⊗`PosInBar`
(8 + `RootIntFiP`⊗`FunctionType`⊗`PosInBar`)

Figure 6.4: Viewpoint selections for multiple viewpoint systems predicting `Root`⊗`ChordType`. Viewpoint selection is run on an unrestricted viewpoint pool (triangles), and a restricted viewpoint pool excluding `ChordType` (squares), both with an unweighted Ψ'. Viewpoints added at each iteration are shown below the graph, viewpoints added after selection has been curtailed are parenthesised and plotted with a dashed line.

6.5 Conclusions and Discussion

This chapter has presented a new method for improving predictions from derived viewpoints by weighting Ψ' (the function that maps from the derived to basic alphabets of viewpoints) with the zero-order frequencies of the basic attribute. Results show that such a weighting significantly improves the performance of derived viewpoints that abstract heavily away from their basic viewpoint, notably `MajType`, `7Type`, and `FunctionType`. On the other hand, viewpoints derived from `Root`, such as `RootInt`, `MeeusInt`, and `ChromaDist`, see only marginal improvements or slight decreases in performance. §6.2 posed the idea that derived viewpoints perform poorly due to a mixture of two factors. Firstly, they abstract information poorly from the training data; in other words, the structure they are trying to capture is not present to the extent expected. Secondly, information is lost when mapping from derived to basic domains, as derived elements typically have a one-to-many relationship with elements in the basic domain. By weighting the inverse function, this research has specifically attempted to address the second of these factors, whilst keeping the first factor constant (the training data is not altered). The improved performance when weighting the inverse function for individual viewpoints (§6.4.1) suggests, that to a large extent, the second factor is at play for viewpoints derived from `ChordType`. However, the fact that none of the derived viewpoints outperform their basic viewpoint after weighting the inverse function suggests that the first factor accounts for a sizeable loss of the predictive performance. In other words, the structure anticipated in the corpus for the derived viewpoints to exploit is not as prominent as expected.

It is initially surprising that weighting the inverse function did not allow viewpoints derived from `ChordType` to be selected with the viewpoint selection algorithm (§6.4.2). However, since the individual derived viewpoints do not outperform their basic counterparts (see §6.4.1), they are unlikely to be selected with a greedy search algorithm. Given that they are not present in any linked viewpoints in the viewpoint selection it can be assumed that this result holds for the relevant linked viewpoints tested during viewpoint selection. The viewpoints selected confirm the findings of §5.7, which studied viewpoint selection and bias combination optimisation in the context of merged attributes. §5.7 ran viewpoint selection first on assumed (from Pearce, 2005) LTM-STM and viewpoint combination biases, before optimizing the biases themselves. It is feasible that the new optimal bias would result in different viewpoints selected. However, as the viewpoint selection results in §6.4.2 show, identical viewpoints are selected at each iteration, adding strength to the validity of the combination biases originally found in §5.7.

A final finding of the chapter was that using a weighted inverse function allows highly compact (in terms of the domain sizes of the predictive viewpoints) viewpoint models to be constructed with a relatively small loss in information content. If an unweighted inverse function is used the loss in performance is considerably greater, arguably too much to justify the more compact multiple viewpoint system. From a computational perspective, using compact multiple viewpoint systems are more efficient in time and space, so weighting the inverse function may have useful task-specific applications. From a cognitive perspective compact representations of music are important for the efficient processing of information. It is likely, although not specifically studied to date, that humans accept a slight loss of information in favour of a more compact representation (e.g. pitch intervals). Whilst the current research cannot comment directly on cognitive processes carried out in the mind, it does show that a computational model of cognition is able to exhibit this behaviour. Derived viewpoints such as `FunctionType` and `MajType` are highly compact predictors of `ChordType`, at the price of a small loss in performance (providing the inverse function is weighted).

If a slight loss of performance in favour of compact viewpoint systems is accepted, the weighted Ψ' model constructs more convincing viewpoint systems from a musicological perspective. Chord function is an important aspect of jazz music (Levine, 1995) and tonal harmony in general, where common cadences progress in *pre-dominant, dominant, tonic,* sequences. Therefore, the fact that `FunctionType` is selected over `MajType` and `7Type` suggests that chord function as signified by the 3rd and 7th of the chord together is more important than the quality of the 3rd (modelled by `MajType`) or 7th (modelled by `7type`) separately.

This research studied weighting only by zero-order frequency. Useful future research might explore alternative weighting schemes beyond zero-order frequencies, such as first-order Markov, or even more aggressive, exponential weighting schemes. Furthermore, applying the weighting schemes to a range of domains, genres, and corpora beyond jazz harmony is necessary to prove that the methods presented in this chapter can be universally applied.

Chapter 7

Testing the Optimality of the Viewpoint Selection Algorithm

7.1 Overview

A brief experiment analyses the viewpoint selection algorithm presented in §3.5, and modified in §5.7.1. The viewpoint selection algorithm used in the present research (Algorithm 1) is a greedy forward stepwise algorithm, aiming to find a locally optimal set of viewpoints for a multiple viewpoint system. Viewpoint selection starts from the empty set of viewpoints (both in the current research and in Pearce, 2005), or a set consisting of the basic viewpoints associated with the basic attributes being predicted (Whorley, 2013). Viewpoints deletions and additions are trialled and selected greedily using mean information content ($\bar{h}$) as a heuristic, halting when no deletion or addition results in a reduction in $\bar{h}$ (Pearce, 2005; Whorley, 2013), or, for the current research, when the reduction does not exceed an effect-size threshold. The selected viewpoint system is not guaranteed to represent a global minimum because of the fixed starting point, greedy selection, and lack of backtracking. The additional effect size halting criteria introduced in the current research means that the selected system may not even represent a true local minimum. The purpose of this experiment is to assess whether the multiple viewpoint systems selected by the viewpoint selection algorithm are acceptable local minima. This is achieved by finding multiple local minima by initialising the selection algorithm at random points in the search space and measuring the difference in performance between the resulting viewpoint systems.

Viewpoint selection results from related studies are reviewed in §7.2. The experimental methodology and hypothesis are presented in §7.3 and §7.4 respectively, and the results in §7.5. Finally, some concluding comments are given in §7.6.

7.2 Behaviour of the Viewpoint Selection Algorithm in Related Research

An overview of previous studies using the viewpoint selection algorithm for different corpora and domains is useful at this stage. Pearce (2005, p. 122) proposes the viewpoint selection algorithm as an objective, empirical method for constructing multiple viewpoint systems; previously Conklin and Witten (1995) selected viewpoints by hand using expert knowledge. The viewpoint selection algorithm was tested with a 10-fold cross validation of set of 185 Bach Chorale melodies (dataset 3, Table 4-E), predicting `Pitch` with a LTM+C*I—STMX*UI model (see §5.2.1.4) with combination biases of 7 and 2 for the LTM-STM and viewpoint combinations respectively. The system of nine viewpoints selected (Pearce, 2005, p. 127) outperformed the hand selected viewpoint system of five viewpoints by Conklin and Witten (1995) (1.953 bit/symbol compared to 2.045 bits/symbol). Each iteration of viewpoint selection added a viewpoint, there were no deletions. The system predominantly comprised of linked viewpoints combining various relative pitch and duration representations, and threaded viewpoints representing pitch intervals at the beat and phrase levels in the metrical hierarchy. Interestingly, the first viewpoint selected is pitch interval linked with duration. Noting that the search algorithm is greedy and starts from the empty set, this viewpoint is, therefore, the single most successful predictive viewpoint for the corpus. This, alongside the dominance of viewpoints linking pitch and duration in the selection, suggests correlations between pitch and rhythmic structure in the corpus. Only one viewpoint using absolute pitch is selected suggesting relative pitch structure is a more important representational tool than absolute pitch structure, a perspective reinforced by studies of melodic perception (Saffran & Griepentrog, 2001).

Whorley (2013), in applying multiple viewpoint systems to four-part harmonisations, proposes several variations and modifications to the viewpoint selection algorithm. The modifications are necessary owing to the exponentially larger pool of viewpoints resulting from the expansion from monophonic to polyphonic music. Not only are the number of primitive viewpoints multiplied by the number of voices, but the linking of primitive viewpoints both within and between voices results in the exponential growth of the viewpoint pool. Variations of the algorithm restrict the viewpoint pool in different ways (see Whorley, 2013, p. 124), the overriding principle being that linked viewpoints are only included in the pool if they include a primitive viewpoint already in the pool. All basic viewpoints are automatically included in the pool. Proposed variants of the algorithm include retaining primitive viewpoints for the purposes of defining valid linked viewpoints even after they have potentially been removed as a result of viewpoint deletion, and predefining specific primitive viewpoints that may always be linked to, even if not currently in the viewpoint system. Viewpoints are removed completely from the pool if it is found at any iteration that they increase the mean information content by a certain threshold. As multiple basic attributes are being predicted, viewpoint selection starts from the set of basic viewpoints capable of predicting the relevant basic attributes, rather than the empty set.[1]

Whorley (2013) tests a large number of viewpoint selection runs with various prediction tasks (predicting various numbers of voices in differing orders and combinations), models (STM, LTM, LTM+, BOTH, and BOTH+), and corpora. A selection of interest are reviewed here. A preliminary test selects viewpoints for the pitch and duration attributes of the monophonic melodies from a corpus of 100 hymns (Whorley, 2013, p. 190). The viewpoint selection algorithm runs for 34 iterations, two of which are deletion stages to remove the basic viewpoints representing duration (at iteration 6) and absolute pitch (at iteration 8). Confirming the findings of Pearce (2005), linked and threaded viewpoints dominate the selected viewpoint system. The first four or so viewpoints selected are responsible for the largest reductions in mean information content (referred to as cross-entropy). The first two of these link scale degree and pitch interval viewpoints with phrase structure, and the fourth, scale degree with metre. The third assists the prediction of duration by linking duration and metre. Overall, Pearce's (2005) observation that the most successful linked viewpoints capture both pitch and rhythmical/metrical structure is upheld. It is discovered that when predicting multiple attributes mean information content is reduced if viewpoint selection is run on each attribute separately (Whorley, 2013, ch. 6). When predicting pitch alone for multiple voices (e.g., Whorley, 2013, pp. 236-237) the linked viewpoints typically use scale degree or interval representations linked with viewpoints representing rhythmic, metrical, or phrase structure. When dividing the task of predicting multiple voices into sub-tasks (e.g. bass given soprano, or alto and tenor given soprano and bass) there is little overlap in the selected viewpoints between sub-tasks. In almost all cases for such sub-tasks, viewpoints are selected that link viewpoints in the given voice with viewpoints in the voice to be predicted.

[1] The present research (§5.7.1) also predicts multiple attributes, but simply checks that the viewpoint system being tested at any round of addition or deletion is capable of predicting all of the required basic attributes before calculations are made. This allows the present research to start from the empty set of viewpoints whilst predicting multiple basic attributes.

Pearce and Wiggins (2006)[2] apply the viewpoint selection algorithm to building multiple viewpoint systems that correlate with perceptual studies of melodic prediction in the context of single intervals (Cuddy & Lunney, 1995), British folk songs (Schellenberg, 1996), and chorale melodies (Manzara et al., 1992). Regression coefficients with the goodness-of-fit and mean entropy estimates from the perceptual studies were used as the heuristic to guide viewpoint selection instead of mean information content. For the context consisting of single intervals, viewpoint selection first added a viewpoint linking interval and duration (drawing similarities with Pearce, 2005, p. 127) before adding a viewpoint representing the interval between the current event and the first pitch in the piece, and another viewpoint representing interval class. For the context of British folk melody extracts, several viewpoints capturing relational pitch structure are selected. Interestingly, absolute pitch is selected first, but dropped after a linked viewpoint comprising absolute pitch and inter-onset-interval is selected, suggesting that pitch alone is an inadequate predictor of pitch expectation and relies of rhythmic structure. For the context of Bach chorale melodies, the relationship between the heuristics of mean information content and regression coefficient with participant's entropy estimates is investigated by running viewpoint selections on each heuristic in turn. As expected, there is a clear inverse relationship between the heuristics: at each stage an increase in the regression coefficient corresponds with a decrease in mean information content regardless of which is used as the heuristic. However, there is little overlap between the viewpoint systems selected.

7.3 Experimental Methodology

In order to assess the validity of the viewpoint selection algorithm in terms of selecting reasonable local minima the following methodology is proposed. A typical viewpoint selection algorithm is required to find a reasonable solution from a search space of $2^{105} \approx 4.1 \times 10^{31}$ different viewpoint systems.[3] Many of the viewpoint systems in the space will contain a large number of viewpoints resulting in excessively long run times (see §3.6) and will be inefficient as cognitive models in terms of model compactness (see §6.4.3). Therefore, the current research focuses on the search area containing viewpoint systems of around five or less viewpoints, which is consistent with the viewpoint systems selected in §5.7 and §6.4.

The principle behind the experiment is to force the selection algorithm to search

[2] See also Pearce (2005, ch. 8).

[3] For a typical representation comprising 14 viewpoints, allowing links between any two primitive viewpoints (see §3.5).

areas of the search space that would not be searched if the algorithm had been initialised from the empty set of viewpoints. If the selection algorithm works successfully, and the landscape of the search space is as expected, the addition and deletion steps will allow the process to converge on optimal points in the search space. Ten viewpoint systems, each consisting of five randomly selected viewpoints, are chosen as the initial viewpoint sets for the viewpoint selection algorithm. Each is run as described by Algorithm 1 until termination, which may be curtailed when the model improvement has a Cohen's effect size of $d < 0.005$.

The experiment predicts `Root`$\otimes$`ChordType` across the *Real Book Vol. 1* corpus (dataset 1, Table 4-E). The model used is the best model found in Chapter 6, a STMC*IUM-LTM+C*IM model using bias weights of $b = 2$ and $b = 1$ for LTM-STM and viewpoint combination respectively, weighting Ψ' with zero-order counts as described in §6.3.

7.4 Hypothesis

It is anticipated that the local minima selected by the viewpoint selection algorithm starting from the empty set will be reasonably representative of the search space. Therefore, it is hypothesised that viewpoint selection runs starting from random initial sets of viewpoints will mostly converge to this point. The early curtailing of viewpoint selection may result in termination at nearby points, both in terms of $\bar{h}$ and in terms of similarity between viewpoint systems. If termination occurs at different viewpoint systems it is hypothesised that the difference in performance (as measured by $\bar{h}$) between them will be negligible.

7.5 Results

The viewpoint systems selected from the ten randomly-initialised viewpoint selection runs are summarised in Table 7-A. The ten runs converge on six different viewpoint systems, four of which are arrived at by two selection runs each, and the other two once only. The performance difference ($\bar{h}$) between the six selected viewpoint systems is almost negligible. Although the difference between the best (2.962 bits/symbol) and worst (2.979 bits/symbol) performing systems is found to be statistically significant with a two-sided t-test ($df = 347, t = 5.680, p < 0.001$), the absolute difference of 0.017 bits/symbol and effect size of Cohen's $d = 0.018$ indicate that the difference is unlikely to be substantial in practical terms.

The six different viewpoint systems selected have clear similarities, with only very slight (arguably unimportant) variations between them. All but one of the selected systems include in their linked viewpoints Root, RootInt, and RootIntFiP. One exception (which was arrived at twice) includes ChromaDistFiP instead of RootIntFiP, although this is still a viewpoint that models the relation between the current Root and the first Root of the piece, albeit in a form abstracted using the chroma distance. ChordType is almost exclusively used over its derived viewpoints, despite Ψ' being weighted to assist the predictive power of derived viewpoints. Only one exception selects a linked viewpoint containing FunctionType instead. There is no clear pattern for selecting PosInBar, suggesting possibly that it has only a limited influence on harmonic prediction.

Table 7-A: Multiple viewpoint systems selected from 10 viewpoint selection runs starting from random viewpoint systems.

Viewpoint System	$\bar{h}$	Number of times selected
Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType⊗PosInBar	2.962	2
Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType	2.967	2
Root⊗ChordType⊗ Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗FunctionType⊗PosInBar RootIntFiP⊗ChordType⊗PosInBar	2.969	1
Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗ChordType⊗PosInBar ChromaDistFiP⊗ChordType⊗PosInBar	2.977	2
Root⊗ChordType⊗PosInBar RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType	2.978	2
Root⊗ChordType RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType⊗PosInBar	2.979	1

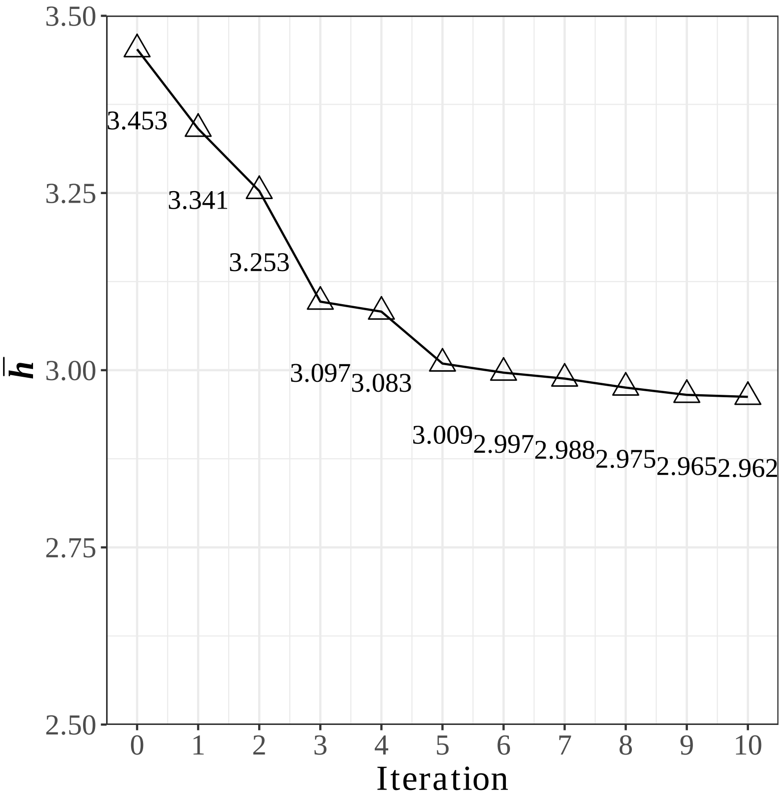
Appendix D shows the deletion/addition rounds of all ten viewpoint selection runs. Figure 7.1 shows a typical run from one of the ten viewpoint selection runs in more detail. The early iterations are mostly deletions, first removing viewpoints containing `MeeusInt`, indicating it is a poor predictor of `Root`. Viewpoints containing `Root` and `RootInt` are added relatively early in the process, both linked with `ChordType` and `PosInBar`, mimicking the first stages of viewpoint selection from the empty set. `ChromaDist`$\otimes$`ChordType` is removed next, followed by the remaining additions. Interestingly, `RootIntFiP`$\otimes$`FunctionType` is not removed until after `RootIntFiP`$\otimes$`ChordType`$\otimes$`PosInBar` has been added, suggesting that whilst it did not reduce the predictive power of the model, it was redundant after the more useful `RootIntFiP`$\otimes$`ChordType`$\otimes$`PosInBar` had been added.

7.6 Conclusion and Discussion

This chapter has assessed the optimality of the viewpoint selection algorithm used to automatically construct multiple viewpoint systems in an information theoretically efficient manner. The established method for initialising the algorithm is to start with the empty set (Pearce, 2005), or a set of viewpoints associated with the basic attributes being predicted (Whorley, 2013). To test whether the locally optimal viewpoint systems constructed using such initialisation techniques are reasonable, other local optima are explored by initialising the selection algorithm from random points in the search space. Results indicate that all random initialisations end in very closely related local minima with mean information content values very close to one another (a range of 0.017 bits/symbol), and with viewpoint systems that are highly related in terms of their linked viewpoints.

An important point to note is that the viewpoint system selected with the lowest $\bar{h}$ (2.962 bits/symbol) is identical to the best performing viewpoint system found so far in Chapter 6 (see Figure 6.2). This is a strong indication that the viewpoint selection algorithm starting from the empty set finds very reasonable local minima, likely global in the restricted search space of viewpoint systems of five viewpoints or fewer.

Future work to further justify the viewpoint selection algorithm would involve repeating this exercise in different genres and domains, and focussing on a larger area of the search space (i.e. more than around five or less viewpoints). The smaller search space is deemed adequate for the current research, which considers both the performance (in terms of mean information content) and compactness (in terms of the total number of symbols in the predictive viewpoint domains, see §6.4.3) of multiple viewpoint systems.



Random initial viewpoint system:
`RootIntFiP`⊗`FunctionType`,
`MeeusInt`⊗`7Type`,
`MeeusInt`⊗`FunctionType`⊗`PosInBar`,
`MeeusInt`⊗`ChordType`⊗`PosInBar`,
`ChromaDist`⊗`ChordType`

Viewpoints selected at each iteration:
1 − `MeeusInt`⊗`7Type`
2 − `MeeusInt`⊗`FunctionType`⊗`PosInBar`
3 + `Root`⊗`ChordType`⊗`PosInBar`
4 − `MeeusInt`⊗`ChordType`⊗`PosInBar`
5 + `RootInt`⊗`ChordType`⊗`PosInBar`
6 − `ChromaDist`⊗`ChordType`
7 + `Root`⊗`ChordType`
8 + `RootInt`⊗`ChordType`
9 + `RootIntFiP`⊗`ChordType`⊗`PosInBar`
10 − `RootIntFiP`⊗`FunctionType`

Figure 7.1: Viewpoint selection for a multiple viewpoint system predicting `Root` ⊗ `ChordType` from an initial set of five random viewpoints.

Chapter 8

The Performance of Absolute and Relative Viewpoints

8.1 Overview

Two types of viewpoints can be used to predict pitch-like features of music, such as note pitches or chord roots. Absolute viewpoints simply represent the pitch sequence with absolute values, whilst relative viewpoints find some relational structure between pitches of two (or, in theory, more) events. If the relationship measured is the distance between adjacent events the viewpoint can be described as intervallic.

This chapter provides a focussed analysis to test the implicit, but strong assumption, established in the multiple viewpoint literature that in general relative viewpoints outperform their absolute counterparts (Conklin & Witten, 1995; Pearce, 2005; Whorley, 2013). Comparisons with equivalent representation schemes in music cognition (§8.2) and computational modelling (§8.3) are discussed, motivating the need for a more in-depth information theoretic analysis. First, the performance of individual absolute and relative viewpoints is compared (§8.4.1). A potential bias concerning predicting the first event of a sequence is addressed in §8.4.2, before the individual relative and absolute viewpoints are tested across a range of order bounds. Finally, the effect of linking relative and absolute viewpoints with temporal viewpoints is observed to reveal any potentially advantageous statistical structure in the correlation between pitch intervals, and rhythmic or metrical structure (§8.4.4). Conclusions are drawn with discussion in §8.5.

8.2 Perceptual and Cognitive Understanding of Absolute and Relative Pitch Representations

A large body of research has been conducted concerning the use of absolute and relative pitch representations in music perception (see McDermott & Oxenham, 2008, for a review). Most humans do not possess a readily accessible absolute representation of pitch (Bachem, 1955; Deutsch, Dooley, Henthorn & Head, 2009; Profita & Bidder, 1988). Absolute pitch (commonly known as perfect pitch) is musically defined as the ability to categorise and label a pitch without context, or to accurately produce a pitch given a note name without context. In cognitive terms, the use of absolute pitch representations may be more implicit, and can be defined as the ability to memorise and recognise sequences of absolute pitch values without prior context. Relative pitch, on the other hand, requires a context for pitch processing tasks, which may be a neighbouring context (in the case of pitch interval), or tonal context (in the case of scale degree). Relative pitch is deemed to be a common perceptual ability, as exemplified by the findings that melodies are in general recognisable when transposed for adults (Attneave & Olson, 1971), as well as for 6-month old infants (Plantinga & Trainor, 2005). The task can be difficult when contour and key are preserved in stimuli (Dowling, 1978; Dowling & Fujitani, 1971), or for atonal melodies if contour only is preserved (Dowling, 1978), both of which are also true for long-term memory (Dowling & Bartlett, 1981). These findings suggest that separate representations at the interval and contour level are at play when processing pitch sequences.

There is evidence to suggest that young infants (six months or less) are capable of absolute pitch recognition (Saffran & Griepentrog, 2001; Volkova, Trehub & Schellenberg, 2006), although melodic structure may have a role in whether absolute or relative representations are used (Saffran, Reeck, Niebuhr & Wilson, 2005). These findings support an *unlearning* theory (Ward & Burns, 1982) of absolute and relative pitch representation, whereby absolute pitch is universal at birth, but usually becomes redundant as an individual develops depending on their environment. Alternatively, an *early learning* theory points to the strong association between musical training at an early age and absolute pitch accuracy (Crozier, 1997; Miyazaki, 1988). Further findings (Baharloo, Johnston, Service, Gitschier & Freimer, 1998; Theusch & Gitschier, 2011) suggest genetic inheritance as a strong factor of absolute pitch ability. Nevertheless, it is apparent that there is a statistical learning element to absolute pitch ability (in terms of both recall speed and accuracy). Simpson and Huron (1994) show reaction time correlates with information content as calculated with a uni-gram statistical model of a large corpus, Miyazaki (1989, 1990) show absolute pitch performance correlates with white piano key notes in the middle of the typical pitch range, and Sergeant (1969) notes absolute pitch ability is enhanced when the stimuli are performed on the participant's main instrument. These studies suggest absolute pitch ability correlates strongly with frequency of exposure, supporting the motivation for statistically-driven computational models of cognition, such as those found throughout the present research. However, it would be a misconception to assume that absolute pitch representation is entirely lost for the majority of the non-infant population; the correct starting notes of familiar songs can be produced or identified with a reasonable margin of error by the majority of people (Levitin, 1994; Schellenberg & Trehub, 2003), with no differences between Asian and Western cultures (Schellenberg & Trehub, 2008). However, the broad pitch range allowed for accuracy of up or down 1 to 2 semitones (Levitin, 1994) does not necessarily imply an absolute pitch representation capable of accurately categorising note names akin to the symbolic representations of pitch in multiple viewpoint systems.

When constructing multiple viewpoint models that correlate with behavioural expectation data in melodies, Pearce and Wiggins (2006)[1] find viewpoints modelling relative pitch (e.g. `cpint`, `cpintfip`, `cpintref`) are important both in correlating with listeners' goodness-of-fit or continuation ratings (corresponding with unexpectedness) for a note given a context, and decreasing mean information content. The relative absence of absolute pitch viewpoints suggests listeners use relative pitch representations when processing musical expectation.

To summarise, relative pitch representations are almost universal among adults, whilst absolute pitch is retained or acquired by a few, possibly through a combination of statistical exposure, genetic predisposition, and early training.

8.3 Absolute and Relative Pitch Representation in Computational Models of Music

The treatment of absolute and relative pitch representations in the literature of computational models for music is more straightforward, although arguably less nuanced. In general, it is taken as given when, for example, undertaking machine learning tasks, that transposed melodies are equivalent. In other words, relative pitch representations are assumed over absolute pitch representations.

A number of techniques are employed to eliminate absolute pitch representations from training corpora. If the key and tonal centre of all pieces are known or can be reliably

[1] See also Pearce (2005, ch. 8).

inferred, each piece can be transposed into the same key, typically C (e.g., Perez-Sancho et al., 2009; Rohrmeier & Cross, 2008). If not, the training data can be transposed to all keys so that every transposition is covered (e.g., Hedges et al., 2014; Pachet, 2012; Pachet & Roy, 2011). Finally, pitch interval (rather than pitch) representations will represent two sequences, one of which is a chromatic transposition of the other, with the same sequence of interval symbols (e.g., Abdallah, Gold & Marsden, 2015; Conklin, 2010; Conklin & Witten, 1995; Marsden, 2005; Temperley, 2014).

Whilst the musicological justification for these techniques is a logical approximation, it does not necessarily follow that useful computational models of music cannot be constructed without them. The distribution of keys, and therefore also pitches, is rarely uniform across a training corpus. The main dataset of the current research, the *Real Book Vol. 1*, has a high frequency of C, F, G, and D chord roots (Figure 4.3), suggesting there is an underlying non-uniform distribution of keys. It follows that useful statistics can be obtained by simply representing pitch (or chord roots) in an absolute fashion. An advantage of multiple viewpoint systems over similar statistical learning approaches is that they are capable of representing and combining models from both forms of representation.

8.4 Analysis of Absolute and Relative Viewpoints in Viewpoint Systems

Given the preference for relative over absolute pitch representations in music perception, a seemingly straightforward hypothesis would be that computational models of music cognition and perception such as IDyOM (Pearce, 2005; Pearce & Wiggins, 2012, see also Chapters 5, 6, and 7) exhibit the same behaviour. The advantage of such explanatory models of cognition and perception is that these kinds of hypotheses can be verified empirically. Specifically, it is expected that this behaviour would be displayed in the viewpoint selection algorithm of IDyOM (see §3.5) by first selecting relative (e.g. `PitchInt`,[2] `RootInt`) over absolute (e.g. `Pitch`, `Root`) viewpoints. Given that it is incredibly rare (see Pearce, 2005; Whorley, 2013, and Chapters 5 and 6 of the current thesis) for a selected viewpoint to be removed during a viewpoint selection process that has been initialised from the empty set of viewpoints on an unconstrained viewpoint pool (see Whorley, 2013), the first viewpoint selected represents the single viewpoint with the highest individual performance (i.e. the lowest mean information content, or the viewpoint which best encodes the test data). With this in mind, the current chapter

[2] Equivalent to `cpint` (Pearce, 2005, p. 60).

limits its scope to that of single viewpoint systems in order to observe the performance of relative viewpoints in detail across a range of corpora and domains.

The hypothesis that relative viewpoints outperform absolute viewpoints is partly borne out in the viewpoint selection results of Pearce (2005, pp. 127, 172) and Whorley (2013, p. 190), where the first viewpoints objectively selected for different experiments are `cpint`$\otimes$`dur`, `cpintfref`$\otimes$`cpint`, and `ScaleDegree`$\otimes$`Phrase`. Indeed, each linked viewpoint contains a relative pitch representation; absolute pitch representations are not selected. However, the linked viewpoints obscure the performance of individual viewpoints as they are finding statistical structure in the correlations between basic attributes. It does not follow that because `PitchInt`$\otimes$`Duration` outperforms `Pitch`, `PitchInt` alone also outperforms `Pitch`.

Interestingly, the viewpoint selection results in the current research (§5.7, §6.4.2, and §7.5) appear not to conform to the assumption that relative viewpoints outperform absolute when applied in the harmonic domain. `Root`$\otimes$`ChordType`$\otimes$`PosInBar` is consistently selected first instead of, for example, `RootInt`$\otimes$`ChordType`$\otimes$`PosInBar`. When initialised with random viewpoint systems for viewpoint selection the largest drop in mean information content coincides with the addition of a linked viewpoint containing `Root` rather than `RootInt` (e.g. Figure 7.1).

In the current chapter, a number of analyses are conducted to provide a deeper understanding of this phenomenon. §8.4.1 compares the performance of primitive (non-linked) absolute and relative viewpoints for five datasets across melodic and harmonic domains. §8.4.2 tests whether relative viewpoints suffer significantly as they are unable to predict the first event of a sequence. §8.4.3 observes the effect of restricting the order-bounds of models using relative and absolute viewpoints. Finally, §8.4.4 explores the impact of correlations between relative viewpoints and temporal structure on model performance.

8.4.1 Basic Performance of Primitive Viewpoints

First, the performance of primitive absolute and relative viewpoints is assessed for the five datasets used in the current research (Table 4-E). For the melodic datasets (3, 4, and 5) `Pitch` is predicted either with the absolute viewpoint `Pitch` or the relative viewpoint `PitchInt`. Similarly, for the harmonic datasets (1 and 2) `Root` is predicted with the absolute viewpoint `Root` or the relative viewpoint `RootInt`. Close to optimal parameterisations are used for models predicting each dataset. For the melodic datasets, an STMX*UI-LTM+C*I (see §5.2.1.4 for model notation shorthand) uses a bias of 7 for LTM-STM combination (Pearce, 2005, ch. 6, 7). For dataset 1, an STMD*IU-LTM+C*I uses a bias of 2 for LTM-STM (Tables 5-B, and 5-D). Similarly,[3] for dataset 2, an STMD*I-LTM+C*I uses a bias of 2 for LTM-STM combination (Tables 5-B, and 5-D). All probability distributions are combined with a weighted geometric combination scheme. Ψ' was left unweighted for all models as it was found to have only a minimal impact on viewpoints derived from `Root` in Chapter 6. The performance of the relative viewpoint is compared with the absolute for each dataset with the expectation that relative viewpoints capturing intervallic structure outperform their basic counterparts.

The results (Table 8-A) universally defy the loose hypothesis that primitive relative viewpoints outperform their absolute counterparts in single viewpoint systems. For all but dataset 1, the difference produces a notable but small effect size of around $0.2-0.3$ in favour of absolute viewpoints. §5.7.4 mused that the performance of `Root` over `RootInt` may, in part, be because of the small domain of `Root` reducing sparsity problems, coupled with clear statistical structure in the zero-order distributions (Figure 4.3). However, this argument is unconvincing considering the results are replicated for `Pitch` and `PitchInt`, which have large domains and are more likely to suffer from sparsity issues. Initial findings, therefore, find no information-theoretic motivation for the preference of relative of absolute representations.

Table 8-A: The performance ($\bar{h}$) of primitive absolute and relative viewpoints predicting `Root` for datasets 1 and 2, and `Pitch` for datasets 3, 4, and 5.

Dataset ID	Absolute viewpoint		Relative viewpoint		Cohen's d
	τ	$\bar{h}$	τ	$\bar{h}$	
1	`Root`	2.252	`RootInt`	2.289	-0.080
2	`Root`	1.377	`RootInt`	1.491	-0.268
3	`Pitch`	2.300	`PitchInt`	2.415	-0.237
4	`Pitch`	2.571	`PitchInt`	2.744	-0.350
5	`Pitch`	2.473	`PitchInt`	2.620	-0.312

Note. No differences are significant (after Bonferroni correction) at the $p < 0.001$ level as judged by a one-sided t-test over pieces.

8.4.2 Predicting the First Event of a Sequence

Intervallic viewpoints (such as `RootInt` and `PitchInt`) are unable to predict the first event of a sequence because they lack the necessary previous event to form an interval

[3] Identical apart from the STM not using update exclusion.

(see §4.4.1.2). Therefore, for the first event of a sequence, these viewpoints return the undefined element, $\perp$, and produce a uniform distribution over the domain of the basic attribute being predicted.

The following section tests whether the inability to fully predict the first event of a sequence can account for relative viewpoints surprisingly poor performance in comparison with absolute viewpoints. The experimental methodology is identical to §8.4.1 with the exception that the first prediction of each piece is removed (only) for the purposes of calculating mean information content, $\bar{h}$. By way of comparison, the mean information content is also calculated removing all but the first prediction of each piece. It is hypothesised that the performance difference between relative and absolute viewpoints observed in §8.4.1 is notably reduced, or even reversed. In cognitive terms, this would imply that although absolute pitch representations are possibly used for processing the initial events of sequences, relative pitch representations dominate the subsequent processing.

Table 8-B: The performance ($\bar{h}$) of primitive absolute and relative viewpoints over the first event in each piece (upper half), and over all events except the first event in each piece (lower half).

	Dataset ID	Absolute viewpoint		Relative viewpoint		Cohen's d
		τ	$\bar{h}$	τ	$\bar{h}$	
First	1	`Root`	3.546	`RootInt`	3.700	-0.579*
	2	`Root`	3.495	`RootInt`	3.700	-0.406
	3	`Pitch`	3.675	`PitchInt`	4.392	-1.276*
	4	`Pitch`	4.461	`PitchInt`	5.209	-1.849*
	5	`Pitch`	3.857	`PitchInt`	4.700	-1.588*
Rest	1	`Root`	2.221	`RootInt`	2.257	-0.072
	2	`Root`	1.355	`RootInt`	1.468	-0.265*
	3	`Pitch`	2.272	`PitchInt`	2.375	-0.202
	4	`Pitch`	2.538	`PitchInt`	2.701	-0.321*
	5	`Pitch`	2.448	`PitchInt`	2.582	-0.282*

Note. * marks significant differences (after Bonferroni correction) as judged by a two-sided t-test over pieces at the $p < 0.001$ level.

Surprisingly, the results (Table 8-B) maintain the differences in performance between relative and absolute viewpoints found in §8.4.1. Removing the first prediction from the sequence (the lower half of Table 8-B) results in only slightly smaller effect sizes in comparison with the full sequence (Table 8-A). The difference in $\bar{h}$ for dataset 1 is found not to be statistically significant at the $p < 0.001$ level ($df = 347, t = -2.249, p = 0.025$), showing that the relative viewpoint matches the performance of the absolute viewpoint. This implies that these findings are domain-dependant, and are not necessarily truly general, although clear significant differences are found for three out of the other four datasets, which are both melodic and harmonic. Observing the top half of Table 8-B it is apparent that both relative and absolute viewpoints suffer considerably when attempting to predict the first event of the sequence. As discussed above, relative viewpoints are unable to predict the first event of a sequence because they are undefined, $\perp$, and so return a uniform probability distribution, however, this is not the case for absolute viewpoints. A likely reason for absolute viewpoints also being poor predictors of the first event of a sequence is that they lack a context in which to make informed probability estimates, as the variable order model must back off to the 0^{th} order. In summary, the fact that relative viewpoints are outperformed by absolute viewpoints cannot be accounted for by their inability to predict the first event of a sequence.

8.4.3 Order Bounds

The results of §8.4.1 and §8.4.2 strongly suggest that, in general, when considering primitive viewpoints in single viewpoint systems, relative viewpoints (specifically intervallic viewpoints such as `RootInt` and `PitchInt`) do not outperform their absolute counterparts (`Root` and `Pitch`). This result holds even when the first prediction of a piece is omitted from mean information content calculations.

An interesting property of relative, and specifically intervallic, viewpoints is that their resulting viewpoint sequences are one element shorter than the equivalent absolute viewpoint sequence. For example, for a sequence of roots: $e_1^4 = [C, D, G, C]$, $\Phi_{\texttt{Root}}\left(e_1^4\right) = [0, 2, 7, 0]$, and $\Phi_{\texttt{RootInt}}\left(e_1^4\right) = [2, 7, 7]$. Note that Φ_τ (Equation 3.2) removes undefined ($\perp$) elements from viewpoint sequences. This is an important distinction to make when using a bounded PPM model (see §5.2.1), since a context of n viewpoint elements in an intervallic viewpoint will represent abstracted information from $n+1$ events. On the other hand, n viewpoint elements forming a context in an absolute viewpoint model represents information from precisely n events.

So far, the current research has used only unbounded models in the PPM* framework. This may mask any potential advantages of intervallic over absolute viewpoints resulting from fixed order bounds. In computational terms, searching a suffix tree for a subsequence of length m has linear time complexity: $O(m)$ (Gusfield, 1997). If an order bound of g is enforced[4] this is equivalent to $O(g)$. If no order bound is enforced (as in PPM*) g is, in the worst case, the length of the longest sequence in the training data. Although direct comparisons cannot be made between computational implementations and hypothetical cognitive models, it is assumed that the cognitive demands of storing

[4] Or $n-1$ for n-grams.

and matching subsequences of lengths up to g (typically $g \leq 10$) are substantially less than the demands of storing and matching unbounded length subsequences.

Next, a short analysis is presented comparing the performance of relative and absolute viewpoints across a range of order bounds. `Root`, `RootInt`, `Pitch`, and `PitchInt` viewpoints are tested across all five datasets with various order bounds, g, such that $0 \leq g \leq 10$. Otherwise, the model parametrisations are identical §8.4.1 and §8.4.2.

Figure 8.1 shows the effect of varying the order bound on the mean information content for absolute (`Root` or `Pitch`) and relative (`RootInt` or `PitchInt`) viewpoints across the five datasets. In general, absolute viewpoints still outperform their relative counterparts, even when the context is limited by a low order bound. Notable exceptions can be found for datasets 1, 3, and 4, with an order bound of 0 only. With an order bound of 0, relative viewpoints outperform absolute viewpoints significantly for dataset 1 ($df = 347, t = 16.459, p < 0.001$), dataset 3, ($df = 184, t = 11.415, p < 0.001$), and dataset 4 ($df = 565, t = 5.258, p < 0.001$). However, whilst the differences in mean information content is notable for dataset 1 (0.508 bits/symbol, Cohen's $d = 0.917$), and dataset 2 (0.249 bits/symbol, Cohen's $d = 0.985$), only a negligible difference of 0.033 bits/symbol (Cohen's $d = 0.225$) is found for dataset 4.

In summary, absolute viewpoints maintain an unexpected performance advantage over relative viewpoints, even when an order bound is enforced. The results of this brief analysis show that in specific circumstances relative viewpoints will outperform absolute viewpoints: when the order bound is 0 (a unigram model), but only in certain datasets. This suggests that the comparative performance between these two types of viewpoint is inconsistent for the unigram model, and highly dependant on domain and training data. As such, it is difficult to draw general conclusions.

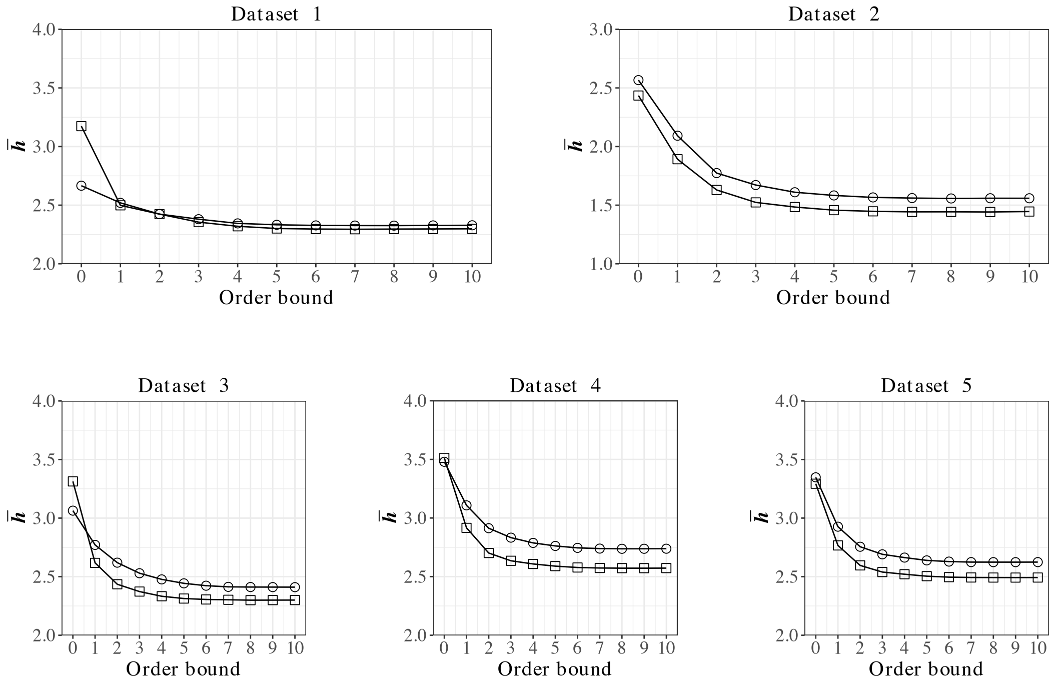


Figure 8.1: Mean information content for relative (circle) and absolute (square) viewpoint models across various order bounds. For datasets 1 and 2 the relative and absolute viewpoints are `RootInt` and `Root` respectively, for datasets 3, 4, and 5 they are `Pitch` and `PitchInt`.

8.4.4 Correlations with Temporal Structure

The overarching conclusion to be drawn from the analyses presented in the preceding sections is that, contrary to expectation, primitive absolute viewpoints such as `Root` and `Pitch` outperform primitive relative viewpoints such as `RootInt` and `PitchInt` in single viewpoint systems. This conclusion holds even when the first event of the sequence is removed from prediction (§8.4.2), and when lower order bounds are enforced (§8.4.3). The implications of these results are that the high performance of linked viewpoints containing primitive relative viewpoints, for example `cpint`$\otimes$`dur` (Pearce, 2005, p. 127), cannot be attributed to the use of the relative viewpoint alone. Instead the high performance can only be attributed to correlating statistical structure, in the case of `cpint`$\otimes$`dur` (Pearce, 2005, p. 127) resulting from `dur`.

This section tests the hypothesis that the performance of relative viewpoints is highly dependant on correlating temporal information. Temporal structure in the current viewpoint representation scheme primarily comprises of duration information (`ICI` and `Duration` for chords, `IOI` and `Duration` for melodies), and metrical structure (`PosInBar`). As the harmonic datasets contain no rests, the `ICI` of an event is always is equivalent to the `Duration` of the previous event, however, this is not the case in the melodic datasets, which do contain rests. Defining `Duration` as a harmonic viewpoint is a departure from the representational formalism presented in §4.4, where `PosInBar` has been deemed sufficient as a basic attribute, with no other basic attributes capturing temporal information (see §4.5). `Duration` is used in the current analysis as a basic harmonic attribute simply so that comparisons can be drawn between equivalent melodic and harmonic viewpoints.

Linked viewpoints comprising relative primitives (`RootInt` or `PitchInt`) and temporal viewpoints (one of `PosInBar`, `ICI`/`IOI`, or `Duration`) are compared with linked viewpoints comprising absolute primitives (`Root` or `Pitch`) and the same temporal viewpoints. The hypothesis tested is that relative viewpoints linked with temporal viewpoints will outperform equivalent linked absolute viewpoints. The model parameters remain unchanged from §8.4.1, with the task of predicting `Root` or `Pitch`. It is worth noting that in order to predict with `PosInBar`, `IOI`, and `Duration` in the melodic dataset the domain of `Onset` must be set dynamically as described by Pearce (2005, p. 65).[5] Note the subtle difference between `Duration` and `ICI`: both are essentially durations, but the former represents the duration of the current chord whilst the latter represents the time interval between the current and *previous* chords.

[5] See also §4.5.

Table 8-C: The performance ($\bar{h}$) of linked absolute and relative viewpoints predicting `Root` for datasets 1 and 2, or `Pitch` for datasets 3, 4, and 5.

Dataset ID	Absolute viewpoint		Relative viewpoint		Cohen's d
	τ	$\bar{h}$	τ	$\bar{h}$	
1	`Root`⊗`PosInBar`	2.193	`RootInt`⊗`PosInBar`	2.199	-0.017
1	`Root`⊗`ICI`	2.272	`RootInt`⊗`ICI`	2.204	0.124*
1	**`Root`⊗`Duration`**	**2.133**	`RootInt`⊗`Duration`	2.159	-0.050
2	`Root`⊗`PosInBar`	1.380	`RootInt`⊗`PosInBar`	1.473	-0.205
2	`Root`⊗`ICI`	1.458	`RootInt`⊗`ICI`	1.505	-0.092
2	**`Root`⊗`Duration`**	**1.346**	`RootInt`⊗`Duration`	1.442	-0.215
3	`Pitch`⊗`PosInBar`	2.426	`PitchInt`⊗`PosInBar`	2.243	0.333*
3	`Pitch`⊗`IOI`	2.424	`PitchInt`⊗`IOI`	2.267	0.317*
3	`Pitch`⊗`Duration`	2.298	**`PitchInt`⊗`Duration`**	**2.193**	0.211
4	`Pitch`⊗`PosInBar`	2.779	`PitchInt`⊗`PosInBar`	2.614	0.266*
4	`Pitch`⊗`IOI`	2.798	`PitchInt`⊗`IOI`	2.655	0.264*
4	`Pitch`⊗`Duration`	2.648	**`PitchInt`⊗`Duration`**	**2.586**	0.122*
5	`Pitch`⊗`PosInBar`	2.664	`PitchInt`⊗`PosInBar`	2.652	0.027
5	`Pitch`⊗`IOI`	2.705	`PitchInt`⊗`IOI`	2.688	0.055
5	**`Pitch`⊗`Duration`**	**2.582**	`PitchInt`⊗`Duration`	2.631	-0.065

Note. * indicates that the relative linked viewpoint significantly outperforms (after Bonferroni correction) the absolute linked viewpoint as judged by a one-sided t-test over pieces at the $p < 0.001$ level. The best performing linked viewpoint for each dataset is underlined in **bold**.

The results, summarised in Table 8-C, show an uneven influence of temporal information on the comparative performances of relative and absolute viewpoints. There is a clear divide between the harmonic and melodic domains. For the harmonic domain (datasets 1 and 2), the addition of temporal information has no overall effect on the performance of relative over absolute viewpoints, with the exception of `ICI` in dataset 1. In the harmonic domain, for datasets 3 and 4 (Bach chorales and German folksongs respectively) linking `PosInBar`, `IOI`, and `Duration` all allow `PitchInt` to outperform `Pitch`. However, for dataset 5 (Canadian folksongs) any positive differences found using `PosInBar` and `IOI` are not statistically significant. It is possible that the differences in relative viewpoint performance between datasets 3 and 4, compared to 1, 2, and 5, can be accounted for by a more even distribution of keys and tonal centres in the Bach chorale (dataset 3) and German folksong (dataset 4) corpora in comparison with the other datasets. The only firm conclusions to be drawn at this point are that the correlation between pitch interval and temporal information is advantageous in the melodic domain, although the effect is highly dependant on the dataset.

8.5 Summary and Conclusions

This chapter has presented an empirical analysis and discussion of the performance of absolute viewpoints (such as `Root` and `Pitch`) and relative viewpoints (such as `RootInt` and `PitchInt`) in single viewpoint systems. Whilst perceptual and cognitive studies (§8.2), computational modelling of music (§8.3), as well as previous multiple viewpoint research (Conklin & Witten, 1995; Pearce, 2005; Whorley, 2013), have highlighted the importance of relative over absolute representations of pitch, this is not borne out in information theoretic terms for individual viewpoints (§8.4.1). The results stand even if allowances are made for relative viewpoints inability to predict the first note of a sequence (§8.4.2), and for short order bounds (§8.4.3) where relative viewpoints in theory carry information from a comparatively longer context of events. However, a correlation between temporal and intervallic information can be exploited for the melodic domain, although this was not found to be universal across all datasets (§8.4.4).

The finding that temporal information is required for relative pitch representations to exhibit an information theoretic advantage over absolute pitch representations is not entirely aligned with behavioural studies in the music perception literature. Many experiments take a reductionist approach and use stimuli without any changes in duration or metrical stress in order to minimise any unintended influences between musical dimensions (e.g., Pearce et al., 2010c; Saffran & Griepentrog, 2001). However, other experiments vary temporal structure either to create more musically realistic stimuli (Volkova et al., 2006), or to induce specific metrical structure to aid prediction timing (Cuddy & Lunney, 1995). Nevertheless, the fact that evidence for strong relative pitch representations can be found without temporal structure indicates that if a purely information theoretic explanation for such a phenomena existed it should be apparent when observing the individual viewpoints as in §8.4.1. This would suggest that an account motivating the use of relative pitch representations purely to reduce information content is somewhat incomplete. A more plausible computational account of these cognitive representations might also take into consideration the compactness of the predictive model, as discussed earlier in the present research (§6.4.3). Certainly, the high performance of relative viewpoints is far more specific than the established multiple viewpoint literature suggests (Conklin & Witten, 1995; Pearce, 2005; Whorley, 2013); as they rely on correlated temporal information and their performance varies across training corpora and domains.

These findings should be viewed in the context of the limitations of the current study, notably that only single viewpoint systems are tested. The results invite further investigation on the performance of relative and absolute viewpoints within full multiple viewpoint systems, in particular, observing whether viewpoints are selected and then subsequently deleted during the greedy stepwise viewpoint selection process. As humans use multiple cognitive representations for pitch (Dowling, 1978; Dowling & Fujitani, 1971), further studies with multiple viewpoint systems are required before making any claims concerning IDyOM's validity as a model of cognitive representations.

Part III

Statistical Learning and Higher Order Structure

Chapter 9

Background: The Information Dynamics of Thinking Model

9.1 Overview, Methodology, and Motivation

The overall goal of Part III of the present research is to develop a partial, exploratory implementation of the Information Dynamics of Thinking (IDyOT) model (Forth et al., 2016; Wiggins, 2012c; Wiggins & Forth, 2015). Within the context of modelling higher order structure in music, IDyOT presents a strongly bottom-up statistically driven method, building hierarchical structure by grouping symbols from the surface layer upwards, producing probabilistic networks capable of accounting for long-term dependencies. The statistical models at the heart of IDyOT are akin to those developed in Part II: using multiple viewpoint representations to make probabilistic predictions with variable order Markov techniques.

IDyOT itself is far more than an account of bottom-up formations of hierarchical structure. The model initially proposed in Wiggins (2012c), and given a formal theoretical description in Wiggins and Forth (2015), is a general cognitive architecture modelling perception, statistical learning, internal cognitive representations, the boundaries of consciousness, and creative behaviour, across multiple domains. The work may be contextualised within the framework of the 'hierarchical prediction machine' approach to cognition by Clark (2013), where top-down and bottom-up probabilistic processes guided by error-minimization heuristics govern both sensory classification and motor response. The level of description of the cognitive architecture is substantially above the neural substrate, occupying a level of symbolic and geometric representations, and high-level mathematical functions. The methodology follows that of Desain et al. (1998) in building an implementable theoretical models of cognitive process, which may be empirically tested by comparing their behaviour to known human behaviours. IDyOT is an explanatory model (Wiggins, 2007), rather than merely descriptive, in the sense that the model provides an underlying mechanism which drives observable behaviour. Parallels can be drawn with a methodological approach in Music Information Retrieval (Sturm, 2013, 2014) where straightforward metrics such as accuracy, recall, or precision, are insufficient in evaluating the quality of a system. Rather, the underlying mechanisms driving a system must be fully understood to determine precisely *why* a system produces a certain output, as well as *what* exactly the system is modelling.

The current chapter gives a theoretical overview of IDyOT, a cognitive architecture built around a Global Workspace (§9.2). Information theory is used to drive statistical predictions in the model (§9.3), which represents concepts and percepts as symbols inhabiting a geometric space (§9.4). Two hand-constructed examples in the domains of music and natural language are presented (§9.5), showing how IDyOT is theoretically capable of accounting for hierarchical structure, long-term dependencies, and ambiguous symbols. The relationship between IDyOT and computational creativity is discussed in §9.6, and how it relates to the current research in §9.7.

9.2 A Global Workspace Theory Account of Consciousness

Wiggins and Forth (2015) posit IDyOT as an implementation of Baars's (1988) Global Workspace Theory. IDyOT is a cognitive architecture providing an explanatory model of, among other phenomena, implicit learning, expectation, consciousness, and spontaneous creativity.[1] The Global Workspace Theory (Baars, 1988) is an account of consciousness and cognition with a similar framework to an AI blackboard system[2] (Corkill, 1991). Broadly, the Global Workspace defines the limits of conscious thought, with the non-conscious mind comprising of a large number of expert generators in a parallel configuration. Whilst generators may always retrieve information from the Global Workspace, only a single generator at any given time may enter and contribute information. Generators compete for access to the Global Workspace, controlled by a loosely described threshold of 'importance.' Generators that become 'coordinated' (make the same predictions) are synchronised, given a greater volume of importance, and may enter the Global

[1] Referred to as "non-conscious creativity" or "inspiration" in earlier publications (Wiggins, 2012c).

[2] Multiple AI agents have access to a shared knowledge base (the blackboard) with which they may retrieve and contribute information. However, communication between agents is strictly constrained, and is limited to occurring via the blackboard.

Workspace if they surpass an *access threshold*. However, from this arises a fundamental flaw in the theory, identified by Baars (1988) as the *Threshold Paradox*. Coordination between generators is only possible through the Global Workspace, but coordination is required for generators to access the Global Workspace. This paradox is sidestepped by Wiggins and Forth (2015) by using an information theoretic mechanism (discussed in detail in §9.3) to arbitrate access to the Global Workspace.

The Global Workspace Theory takes inspiration from Taine's (1871) metaphorical *Theatre of Conciousness*. Here, the mind is conceived of as a theatre of indefinite depth that has a lit, narrow front with room for a single actor. Other actors inhabit the rest of the darkened stage, which continually widens towards the rear. Conscious focus (the attention of the audience) is described by the process of individual actors passing through the lit front of the stage, making their gestures, and leaving, returning to the (non-conscious) background of the stage where unseen developments continue to take place.

Wiggins (2012c) provides an evocative expansion of this metaphor as an *Operatic Chorus of Mind*. The individual players in Taine's metaphor are replaced with choruses (collections of singers), and soloists who linger longer in the spotlight of conciousness than Taine's continuously fleeting actors. The fixed spotlight is replaced with a follow-spot, roving the stage enacting the focus of the audience. A fundamental difference between the metaphors (and corresponding theories of conciousness) is that the individuals making up the chorus collaborate in a dynamic way, with varying degrees of synchronisation between members drawing focus from the audience. At this point it is beneficial to step from the metaphorical realm into a more theoretical one.

IDyOT is comprised of a large number of statistical generators, with competition for access to the Global Workspace quantified by an information-theoretic measure (proposed in Wiggins, 2012c, outlined in §9.3). Noting the evolutionary and statistical learning motivations for the framework outlined in §9.1, the generators are fundamentally statistical, making Markovian predictions from a symbolic context. The necessity to account for ambiguous input and misinterpretations means that a strictly symbolic representation is rejected in favour of a quasi-symbolic representation based on geometrical *conceptual spaces* (Gardenfors, 2000), discussed more fully in §9.4. An individual generator (Figure 9.1) takes input from the external world (perceptual input) and associated static memory, tracking the information content (Equation 9.2) and entropy (Equation 9.3) of events as they are added to a buffer. The buffer is flushed when the generator encounters a high information symbol (defined formally in §9.3.2), the buffered sequence forms a chunk which is sent to the Global Workspace, displacing the chunk currently in the workspace to memory (Figure 9.2). Chunks are stored into memory using inform-

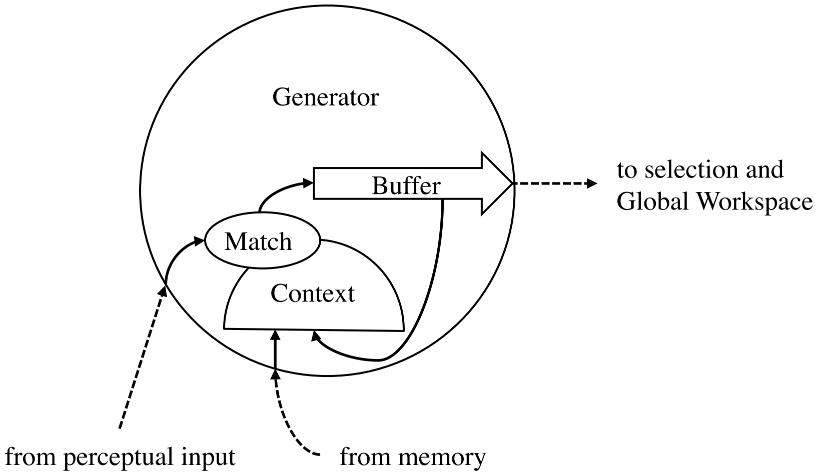


Figure 9.1: An individual generator takes input from the external world and memory, matches events against a probabilistic distribution, and adds them to a buffer. An information theoretic threshold controls when the generator is selected and flushes the buffer to the Global Workspace. Adapted from Wiggins (2012c, Fig. 2).

ation theoretically compact representations and fed back to the underlying statistical models of the generators, completing the cycle of a dynamic system.

It is important to make clear that IDyOT, and indeed Baars's (1988) Global Workspace Theory, is not an attempt of answering the *hard* question (Chalmers, 1996) of what consciousness *is*. Instead, the model is an account of the mechanisms that control which thoughts and percepts come into consciousness, in other words, a theory of *conscious awareness*. In the theatre metaphors the philosophical question of 'who is the audience?' is deferred until the behaviours of the spotlight and audience's attention are understood. In the Global Workspace Theory framework, the core question defines the boundaries of the Global Workspace itself (and thus consciousness) and the information theoretic competition between generators for access.

The following sections provide the necessary detail to complement the outline given above. The nature of the statistical models driving the generators, and the information theoretic mechanism flushing their buffers is discussed in §9.3. Some representational issues arising from purely symbolic representations are addressed in §9.4. §9.5 presents a bottom-up, statistically driven, account of how IDyOT forms hierarchical predictions, capable of capturing higher order structure and long-term dependencies, as well as accounting for cognitive phenomena such as *garden path sentences*. Finally, §9.6 explores a potential explanation of spontaneous creativity, and thus, (weak) *transformational creativity* (Boden, 2003) in the field of computational creativity.

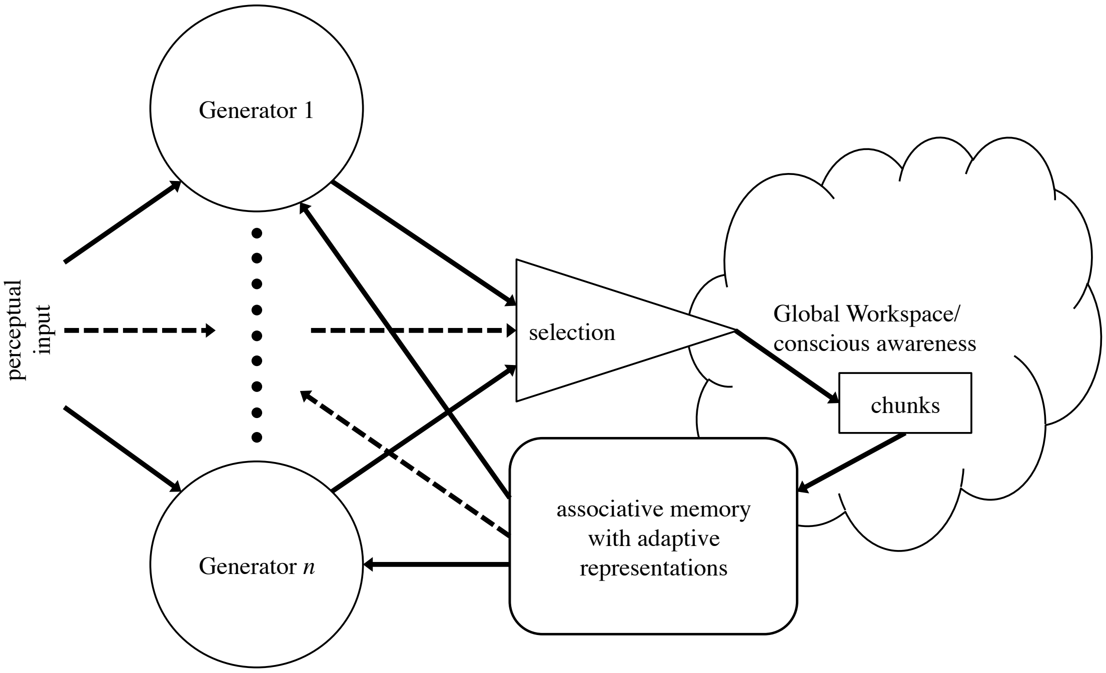

Generator 1
perceptual input
selection
Global Workspace/ conscious awareness
chunks
associative memory with adaptive representations
Generator *n*

Figure 9.2: A schematic description of the IDyOT architecture. Probabilistic Markovian generators (see Figure 9.1) form and flush buffers when selected according to an information theoretic measure of saliency. The Global Workspace holds all chunks, representing them compactly before storing them to memory. Chunks can be fed back to the generators to be added to their probabilistic models. Adapted from Forth, Agres, Purver and Wiggins (2016, Fig. 1).

9.3 Information Theoretic Foundations

The evolutionary motivations (discussed in §9.1, see also §2.2.1, §2.2.2, and Clark, 2013) behind IDyOT establish statistical-driven expectation as the fundamental process of the mind. Expectations from individual generators are governed by statistical, Markovian models (§9.3.1), whilst information theory is used to quantify aspects of expectation, controlling both buffer flushing and entry to the Global Workspace (§9.3.2). By quantifying these key processes in Baars's (1988) Global Workspace Theory, IDyOT presents a computationally implementable, and crucially, empirically testable, model of human consciousness and cognition.

9.3.1 Statistical Underpinnings

The function of the generators in IDyOT (see Figure 9.1) is to continually process information, making predictions of future events given their preceding context with a probabilistic model formed by counting occurrences of previously seen sequences. Therefore, essentially, the generators use Markov models to predict the next symbol, e^i, drawn from a seen alphabet, $\mathcal{A}$, following a context, c, from a probability distribution, $p(e^i|c)$, estimated by frequency counts from data seen so far, $c(e^i|c)$, as in Equation 9.1.

$$p\big(e^i|c\big) \approx \frac{c\big(e^i|c\big)}{\sum_{e\in\mathcal{A}} c(e|c)} \tag{9.1}$$

The input from the outside world is multidimensional in nature; perceived events are not atomic percepts but comprise a number of attributes. Recalling that IDyOT models cognitive processes at an abstracted level above, for example, raw sound waves from speech, a suitable symbolic representation is sufficient for the probabilistic generators. A multiple viewpoint representational framework (Conklin & Witten, 1995) naturally fits these requirements, with each generator taking an individual viewpoint in the LTM or STM memory. Viewpoint representation and modelling schemes are especially appealing as they can be applied to a variety of domains, notably music (Conklin & Witten, 1995; Pearce, 2005), and natural languages (Griffiths et al., 2015; Wiggins, 2012a).

As a developing theoretical model, there is a degree of ambiguity in the precise probabilistic components of the generators themselves which is worth clarifying. Earlier descriptions of IDyOT suggest generators sample from mixed-order, multidimensional, models (Wiggins & Forth, 2015, Fig. 1b) whilst more recent publications suggest sampling from simply first-order, multidimensional models (Forth et al., 2016, Fig. 1). It is anticipated that first-order generators are sufficient (Wiggins, personal communication, July 2017), providing an explanation for mixed-order behaviour arising through higher-order predictions made at levels higher up in the temporal hierarchy (see §9.3.2 and §9.5). Note that although a generator takes input from multidimensional data in memory, they may handle a single dimension akin to an individual viewpoint.

A collection of generators modelling at the level of individual events, can, therefore, be viewed as an IDyOM model as developed by Pearce (2005). The probabilistic models may incorporate improvements relating to predicting multiple voices (Whorley, 2013), correlated attribute prediction (Chapter 5), and derived viewpoints (Chapter 6). To summarise, the generators make predictions from first-order[3] multidimensional Markov models, taking an essentially symbolic multiple viewpoint representation scheme.

[3] Or potentially variable order depending on the details of the implementation (see Wiggins & Forth, 2015, Fig. 1b).

9.3.2 Flushing Chunks and Selection

Generators need not be restricted to generating at the level of individual events. The buffer and flushing mechanism of a generator acts as a chunking device, flushing coherent chunks to the Global Workspace. Adaptive representation schemes may represent these chunks as individual elements, storing them to memory, and returning them to the appropriate generators. Noting that a chunk, originally comprising a sequence of elements, is now represented as a single element (abstracting some information away), generators receiving these chunks from memory are now predicting at a different temporal level to those working on individual events perceived from the external world. §9.5 discusses generators predicting over a hierarchy of temporal levels in more detail. The current section describes the information theoretic mechanism by which chunks are selected and flushed to the Global Workspace.

The principle behind the mechanism is that unexpected or difficult to anticipate events flush the buffer of a generator to the Global Workspace, marking the start of a chunk. The theory has evolutionary and perceptual motivations; there is a distinct evolutionary advantage in being aware of unexpected, potentially threatening events, and it has been shown that people perceive segment boundaries coinciding with low probability events (Pearce et al., 2010b; Saffran et al., 1996; Saffran et al., 1999). Wiggins (2012c) quantifies this process through information theory, from which clear measures of unexpectedness and uncertainty can be derived.

The unexpectedness of an event given a preceding context is given by *information content* (MacKay, 2003, see also §3.4.2); an estimate of the number of bits required to represent a perceived event. Following the notational conventions established in §9.3.1, this is given by the negative log probability:

$$h\big(e^i|c\big) = -\log_2 p(e^i|c). \tag{9.2}$$

Simply put, expected events have a low information content, whilst unexpected events have a high information content. Wiggins (2012c) makes a distinction between *recognition-h* and *prediction-h*. Given a state t currently being perceived, *recognition-h* is simply h_t, or the unexpectedness of the current event, whilst *prediction-h*, or h_{t+1}, is the unexpectedness of a potential following event.

Entropy (MacKay, 2003; Shannon, 1948) is a quantified measure of uncertainty, given a context and probabilistic model, how certain a model or organism is in predicting the following event. Entropy (Equation 9.3) is highest when all probabilities are equal, representing maximal uncertainty, and is zero when a single event is certain.

$$H(c) = \sum_{e \in \mathcal{A}} p(e|c)h(e|c) = -\sum_{e \in \mathcal{A}} p(e|c) \log_2 p(e|c) \tag{9.3}$$

Wiggins (2012c) quantifies a measure of 'audibility' for generators in the Global Workspace, by which generators compete for access to the Global Workspace. Highly expected events do not hold enough information to be salient, whilst highly unexpected events are unlikely to occur (in the *prediction-h* case), or re-occur (in the *recognition-h* case). Noting that information content and probability are inversely related to each other, Wiggins (2012c) proposes mediating between these extremes by multiplying $p(e|c)$ and $h(e|c)$ to give a measure of audibility. Furthermore, more prominence should be given to less uncertain generators,[4] with uncertainty quantified by entropy, $H(c)$. This gives a measure for the 'volume', $T(e|c)$, of a generator (Equation 9.4). In the 'competition for access' paradigm originally presented by Wiggins (2012c) the generator with the highest 'volume' flushes and gains access to the global workspace.

$$T(e|c) = \frac{p(e|c) \times h(e|c)}{H(c)} \tag{9.4}$$

Wiggins (2012c) proposes T as a way to mediate between overly active *prediction-h* cases, where very unlikely anticipated events continuously flush buffers to the Global Workspace inducing a state of perpetual anxiety. However, this problem does not arise when using *recognition-h*, the information content of the currently perceived event. Highly unexpected events that do actually occur (rather than merely being anticipated) should certainly be audible to the Global Workspace and enter consciousness. The precise definition of the measure is seemingly relaxed in subsequent works (Forth et al., 2016; Wiggins & Forth, 2015), although the principle remains that high information content events enable access to the Global Workspace. Alternative, simpler, measures for triggering a buffer flush might simply be the information content of the perceived event, or the entropy of the current context. A threshold may be defined in terms of an absolute value of either of these measures, or in terms of a rise or peak in either (c.f. Pearce et al., 2010b). These chunking measures and threshold mechanisms are defined, compared, and tested empirically in §10.6 of the present research.

The information theoretic approach holds several advantages over the simpler synchronisation and volume approach of Baars (1988). Firstly, it is computationally implementable, and therefore, crucially, empirically testable. Secondly, access to the Global Workspace is controlled by probabilistic measures, rather than frequency-based ones.

[4] In a similar manner to viewpoint model combination; see Conklin and Witten (1995), Pearce (2005) and §3.4.4.

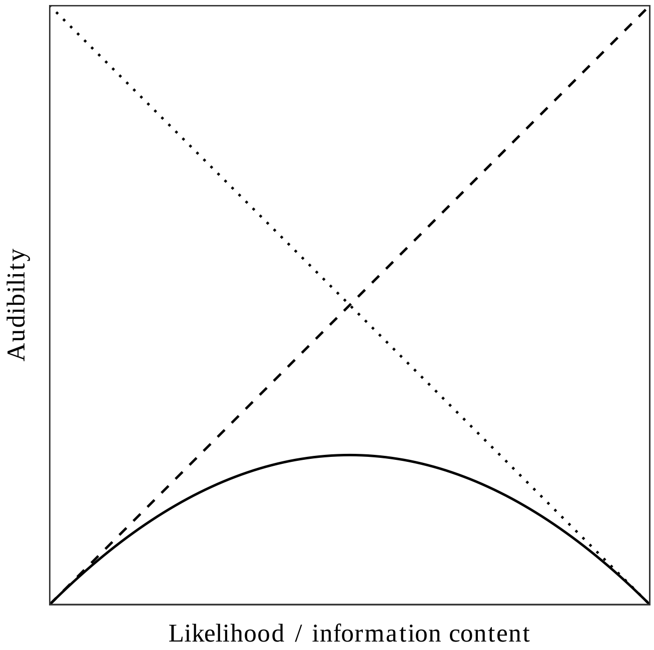


Figure 9.3: 'Audibility' of a generator (solid curve) arising from the interaction between likelihood (p, dashed line) and information content (h, dotted line). Adapted from Wiggins (2012c, Fig. 2). Note that likelihood and information content are plotted on different scales.

Under Baars's theory access to the Global Workspace is gained by a large number of generators predicting in a coordinated manner. An unexpected event will be predicted only by a small number of generators, so will be unlikely to reach the Global Workspace. However, as argued above, quickly becoming conscious to an unexpected event holds notable evolutionary advantages, as well as assisting in the formulation of internal representations. By contrast, with an information theoretic approach a low frequency event will have a low probability and high information content, and thus be more likely to enter the Global Workspace. Finally, the information theoretic approach addresses the threshold paradox, whereby generators surrounding Baars's (1988) Global Workspace are unable to gain access without coordination, and unable to coordinate without access. The approach of Wiggins (2012c) and subsequently Wiggins and Forth (2015) mitigates this issue by removing the need for a definitive threshold. The information theoretic measures of information content and entropy are used to quantify competition between generators with the highest ranked generator gaining access. Implicitly, it is understood that the Global Workspace will, therefore, always hold the buffered input of a single generator at a time.

9.4 Conceptual Spaces and Geometrical Representations

A significant problem arises when representing real world percepts with neatly defined symbolic representations for the purposes of quantifying information theoretic measures. Ordinarily, information theory handles symbols drawn from a finite set acting as the domain,[5] allowing estimates such as entropy to be calculated over a finite distribution. However, unseen symbols outside the domain may be encountered in the real world as novel percepts to an organism. Under a strictly information theoretic approach these new percepts would have a maximal information content (inducing 'unbounded unexpectedness'), and estimates of entropy would become mathematically invalid as the finite alphabet no longer holds. Intuitively, however, humans are not maximally surprised at a novel percept; consider, for example, encountering a novel shade of the colour green that can be readily identified and categorised, or hearing a pitch at the extremes of the range of hearing that hasn't been heard before. In both cases the different shade of green, or a note with the same pitch class, have likely been encountered, suggesting a representation should be capable of accounting for their potential existence before they are specifically encountered. Simply defining a very large finite domain is not sufficient; IDyOT takes a strictly bottom-up approach, avoiding assuming the full set of a domain before it has been encountered. Instead, representations should be learned on the fly, with significant novel encounters of new percepts prompting a re-formalisation of the representation schema in memory.

A secondary issue arises from the fact that atomic symbols are unable to capture the complex multidimensional relations between percepts in cognitive representations. As discussed in §4.4, a multiple viewpoint representation must assume that all symbols in a domain are equally different from one another. In reality, for most domains elements vary in perceptual closeness to one another, with clear similarities and differences evident independent of context.

In light of these issues, *conceptual spaces* (Gardenfors, 2000) have been proposed (Wiggins, 2012c; Wiggins & Forth, 2015) as a valid representational scheme. Conceptual spaces occupy a level of abstraction between the high-dimensional, continuous representations and symbolic representations in a cognitive architecture. They are motivated from the fact that successful organisms must be able to distinguish and categorise events from continuous inputs in an environment, in particular, they are required to accurately place novel events in an existing representation schema. A conceptual space is essentially a geometric space, whose dimensions are perceptual dimensions representing features of objects. Points in the space represent objects, regions represent concepts, and points

[5] By contrast, Ihara (1993) provides an account of information theory in the continuous domain.

near the centre of regions represent prototypes. A *natural category* (Rosch, 1973) is a convex region in the space such that any point lying between any two other points in the space is also in the space. Finally, quality dimensions that are intrinsically linked in a way that an object that can be described by one of them must be described by all of them are referred to as *integral dimensions*.[6] The geometrical nature of the space allows distances between points to be quantified, as well as the categories associated with regions to be mapped onto symbolic representations.

A conceptual space of rhythm and meter (Forth, 2012) is the foundation of a hypothetical account of entrainment in IDyOT (Forth et al., 2016). In addition to predicting *what* an event is, the important concept of *when* a predicted event takes places is addressed. Accurate predictions of when events will occur is important in assisting with the allocation of attention. This can be achieved by predicting time intervals, situated in a geometric conceptual space, given a periodic context. The conceptual space is high-dimensional, capable of representing the periodic component of any well-formed hierarchical metrical structure. The space is abstracted away from specific real time values from performances so that a point in the space may represent several performances of a given rhythm.

In addition to representing individual events as points in a conceptual space, geometric representations can be constructed where sequences of events map onto points. In this case, the dimensions of the space are somewhat arbitrary, but are constructed in such a way that distances (Euclidean or city-block) in the space correspond to perceptual similarity, and such that mean information content of perceived events is minimised. This powerful representation scheme allows chunks in the Global Workspace to be represented, stored in memory, and fed back to the probabilistic generators working at temporal levels above that of individual events.

9.5 Hierarchical Predictions

The above sections have described a collection (which, so far, is unordered) of generators working at different temporal levels, releasing chunks into the Global Workspace according to an information theoretic mechanism. The current section gives structure to the collection of generators, forming a hierarchical network. Hierarchical structure is an implicit structural feature of natural language and music, but is not readily explained with strongly bottom-up statistical (or learned, see §2.3.2) approaches (see Rohrmeier, 2011). On the other hand, fundamentally top-down approaches (or 'non-learned,' see §2.3.1)

[6] For example, hue, saturation and luminance in a conceptual space representing colours.

such as formal grammars account for hierarchical structure, but are poorly cognitively motivated (Wiggins et al., 2010).

IDyOT can account for hierarchical structure, despite the strongly bottom-up statistical approach, by stratifying the collection of generators into temporal layers with the following process (Wiggins & Forth, 2015). Starting at the bottom layer a set of generators process percepts relating to atomic events.[7] The chunks flushed from these generators enter the Global Workspace, are consolidated into memory as symbols mapping onto points in a conceptual space, and returned to a different set of generators, constituting one temporal layer above the bottom. This process is repeated by this set of generators, and the cycle repeats, with generators sending chunks to the temporal layer above via the Global Workspace. In making predictions, generators have access to the current buffers of the layer above. The overall structure at a given point in parsing a sequence might be described as a Bayesian Network. The network is stratified into layers with constraints that conditional dependencies within a layer may only be between adjacent nodes and in a forward direction. Conditional dependencies are additionally permitted between nodes one layer up, or down, from any given node providing they are vertically aligned. Therefore, generators may assist in predicting one symbol forwards, upwards (by flushing), or downwards. An important point to note is that a prediction one symbol forwards in a generator on a given level may refer to symbols on lower levels that are arbitrarily far into the future. This quality, in theory, allows IDyOT to model probabilistically any long-term dependency.

Two specific examples are given below of hierarchical parses in the domains of natural language and Western tonal harmony. Both deal with potentially ambiguous situations that may be accounted for by the parallel nature of IDyOT. The examples are constructed by hand, without a specific implementation, following the model description given in Wiggins and Forth (2015).

9.5.1 Parsing Ambiguous Sentences

The potentially phonetically ambiguous sentence "The horse race passed the barn"[8] is parsed by IDyOT in Figure 9.4. Phonemes[9] are chosen as a suitably low-level cognitive percept for the entry level of the model, whose symbols can be conceived of as regions in a conceptual space. Low probability transitions, and thus high information content

[7] This is the entry point of the model. The perceptual processes involved with, for example, extracting a pitch from a sound wave with complex frequencies, are beyond the scope of the model.

[8] It is worth noting that this sentence is modified from the common *garden path* example: "The horse raced past the barn fell."

[9] Represented in text with the International Phonetic Alphabet for UK Standard English.

symbols, trigger new chunks signifying morpheme groupings. Morphemes are grouped into words, which are subsequently grouped into arbitrary higher level groupings akin to *parts of speech* and syntactic units. Note that from the first phoneme parsed, predictions upwards and forwards begin to generate possible sentence structures.

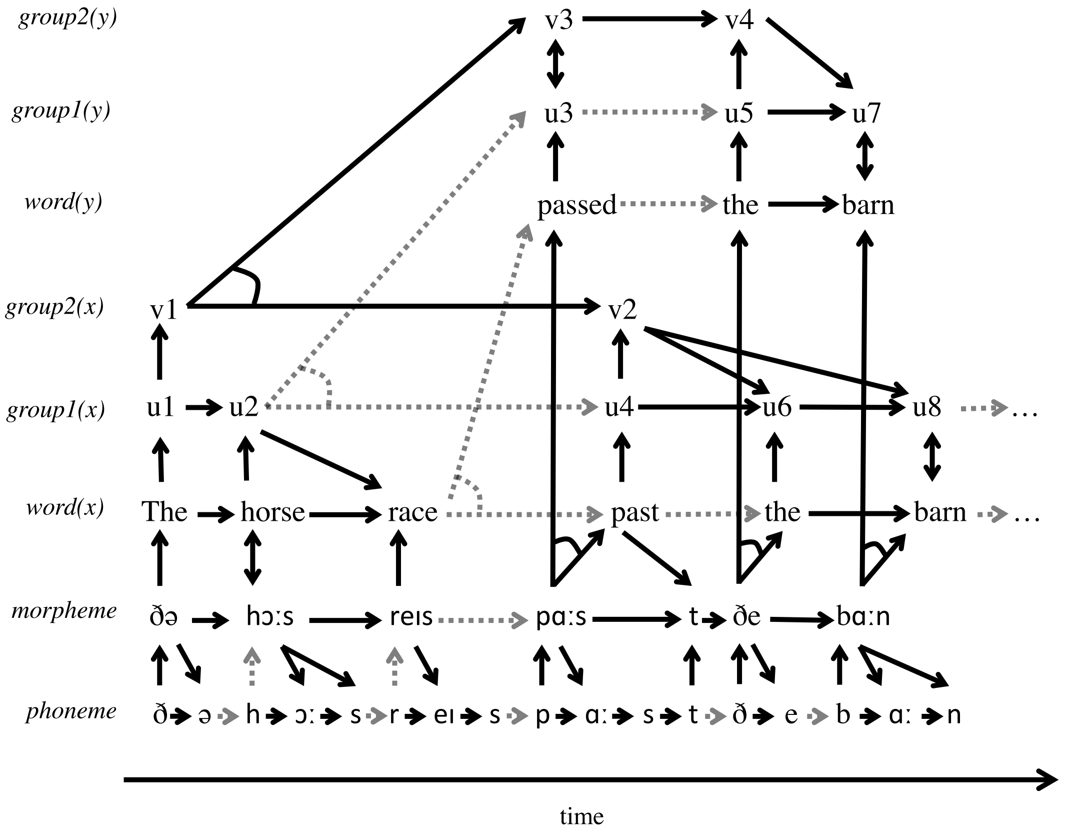


Figure 9.4: Parsing the ambiguous sentence "The horse race passed the barn." The model begins at the phoneme level, chunking to form morphemes, and subsequently words. Higher level groups are somewhat arbitrary, as are their associated symbols. Ambiguous points in the sentence are marked by arcs between edges, splitting the parsing into two parallel streams x, and y. Information flow is roughly indicated by arrow heads, with grey dashed arrows indicating relatively low probability events. Adapted from Wiggins and Forth (2015, Fig. 2).

The crux of the ambiguity is the confusion between 'passed' and 'past,' both of which are represented by the same phonetic symbols: [p aː s t], and thus without context are audibly indistinguishable. When the morpheme group, [paːs], is encountered, upward groupings split into two parallel paths, which could be considered as two groups of generators. Each path is maintained until it becomes highly improbable as further phonemes are parsed. In this case, a long pause after 'barn' suggests an ending to a grammatically coherent sentence containing with verb 'passed,' whereas, a continuation of the sentence (e.g. '...was a roaring success') suggests the adverb 'past' was present.

9.5.2 Parsing Modulations in Western Tonal Harmony[10]

A key concept ensuring smooth modulations[11] in Western tonal music is the use of *pivot chords.* A pivot chord is closely related to both the preceding and following tonal centres simultaneously, so that the boundary between the two is ambiguous and not easily noticeable to a listener. A pivot chord, therefore, provides a useful thought experiment for IDyOT, presented here as another hand-parsed example.

Figure 9.5 shows a short Bach chorale phrase, with the lowest level consisting of chords. Lower levels of sets of notes, and notes themselves are, of course, implicit in the model, but for the purposes of the current research (see §4.2) it is convenient and appropriate from a reductionist perspective to treat chord symbols as a musical surface. It is perfectly plausible for the chunking mechanisms used in IDyOT to also be applied to grouping pitches into chords, but this is beyond the scope of the current example. The phrase begins in A minor (established with a clear imperfect cadence before the extract begins), and modulates to its relative key: C major. The pivot chord, a D minor triad occurring on beat 2 of the 1st bar, is closely related to both tonal centres, as the *iv* of A (the subdominant), and *ii* of C (the supertonic). These two possibilities are held briefly in parallel until it becomes clear that the tonal centre has moved to C owing to high information content symbols being consistently perceived in the old 'A minor' branch (c.f. the Krumhansl-Schmuckler key finding algorithm, Krumhansl, 1990, ch. 4).

[10] Whilst §9.5.1 presented a linguistic example borrowed from Wiggins and Forth (2015), the musical example in this section is novel to the present research.

[11] Moving between closely related tonal centres.

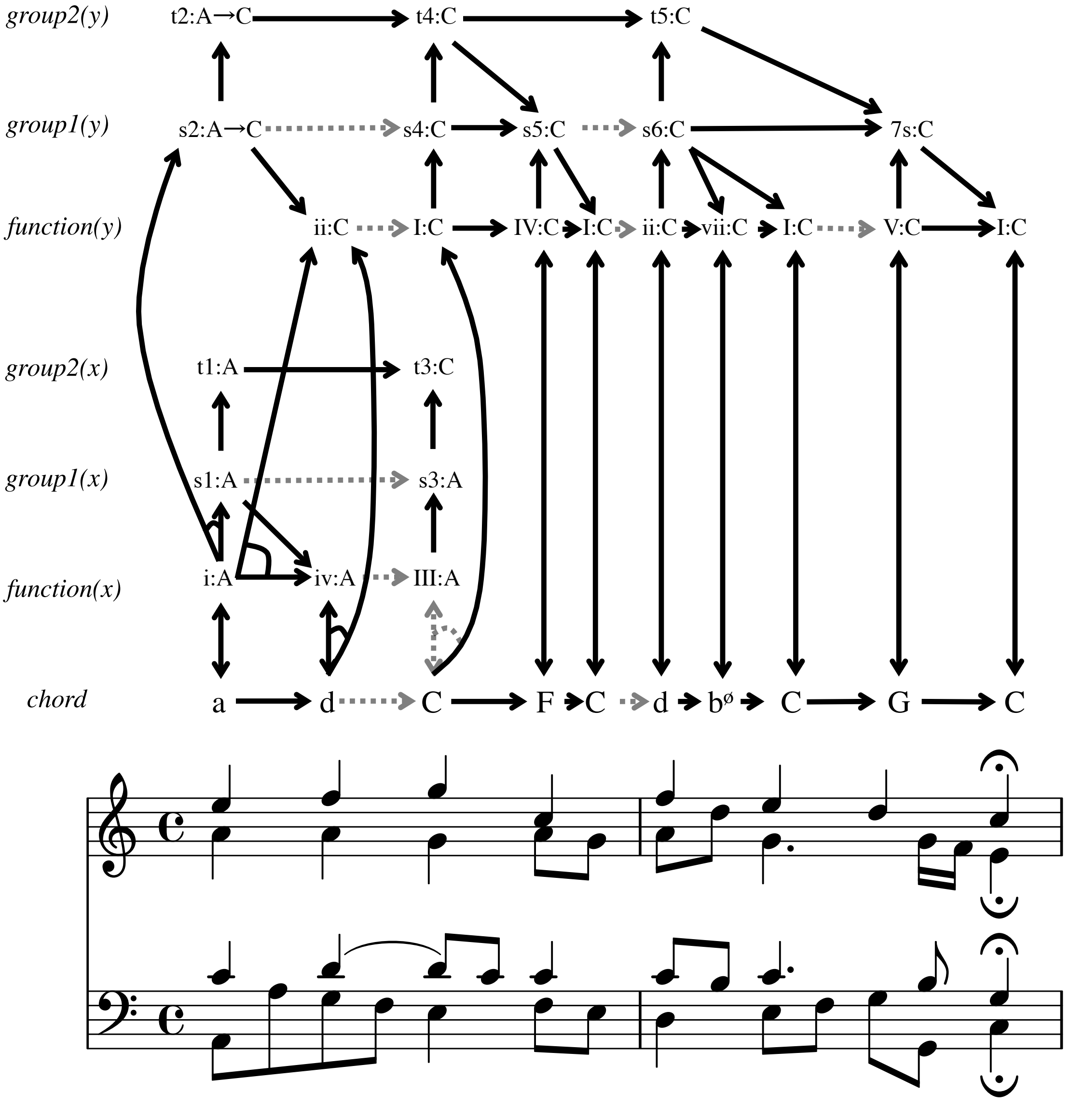


Figure 9.5: Parsing the harmony of bars 5-6 of the Bach chorale *"Herr, ich Habe Missgehandelt"*, BWV 330. Capital letter note names denote major triads, whilst lower case denote minor, and diminished. Symbols above the surface layer are anchored by a tonal centre (denoted by a preceding ':'), which may modulate (denoted by a →). Higher level groups are somewhat arbitrary, as are their associated symbols. Ambiguous points in the phrase are marked by arcs between edges, splitting the parsing into two parallel streams x, and y. Information flow is roughly indicated by arrow heads, with grey dashed arrows indicating relatively low probability events.

A notable difference between the harmonic and linguistic examples is that all symbols above the surface *chord* layer are anchored by local tonal centres. The purpose of the *function* level is simply to assign tonal centres and scale degrees to the chord symbols of the surface layer, no grouping takes place. Higher levels groups may consume multiple tonal centres of symbols at the lower levels, so are permitted to hold several modulating tonal centres (e.g. s2 and t2). In contrast to a linguistic parsing, a parsing of tonal harmonic events is highly relational in a precise, intervallic manner. Tonal centres are used as anchors to predict forwards and downwards, with relative representations abstracting away from absolute ones. These forms of representation (Chew, 2002; Cohn, 1998; Longuet-Higgins, 1978) can easily be accommodated as a conceptual space.

9.6 Computational Creativity

The mechanisms driving prediction, memory, and representation in IDyOT are presented as an explanatory model for both human and computational creative behaviour (Wiggins & Forth, 2015). Particularly relevant to computational creativity, truly creative behaviour cannot merely be the re-categorisation and re-ordering of symbolic input, rather, referential semantics are required to give meaning to symbols. Music does not exhibit semantics referring specifically to things in the external world. Following the works of Meyer (1956), Wiggins (1998), and Huron (2006), musical meaning is distinct from linguistic meaning, and in this context refers to what (for example, emotion) might be communicated through patterns of expectation generated by (statistical) structure. In this sense, a sophisticated statistical model of music cognition and perception can be argued to capture the meaning of music, and thus, may exhibit creative behaviour.

Boden (2003) defines two types of creative behaviour. Firstly, *exploratory creativity* is searching for novel artefacts within a defined conceptual space,[12] and secondly, *transformational creativity* is finding novel artefacts by manipulating the conceptual space itself. Music generation methods using IDyOM exhibit exploratory creativity by sampling from statistical models (Pearce & Wiggins, 2007; Whorley, 2013; Whorley & Conklin, 2016), essentially searching for novel compositions within a defined search space and representation scheme. Wiggins and Forth (2015) argues that the hierarchical structures of IDyOT, allowing representation schemes in the form of conceptual spaces to be built and manipulated on the fly, enable a (weak) form of transformational creativity through statistical learning. Here, statistical learning not only drives the generation, but also the underlying representational schemes. Furthermore, an account of spontaneous

[12] A space of concepts, a different terminology to Gardenfors's (2000) conceptual spaces.

creativity, or more colloquially, an ‘aha!’ moment (Wallace, 1926), can be made by considering the behaviour of the statistical generators without perceptual input. Without perceptual input of the relevant domain, generators are permitted to freewheel, making predictions with input from memory (see Figures 9.1 and 9.2). Chunks from freewheeling generators may enter the Global Workspace, as described in §9.3.2, where they come into conscious thought. This creates an experience akin to suddenly finding a solution to a problem which one has not been consciously solving, or the kernel of a musical idea coming to mind seemingly from nowhere.

9.7 Relation to the Current Research

The level of description of IDyOT presented in this chapter, and by Wiggins and Forth (2015), is that of a general, high-level theory that provides the outline for a model which may be implemented with further detail. To date, a full, computational implementation of IDyOT does not exist; three key areas lack the required level of detail. The first is the information theoretic chunking mechanism, encompassing several techniques by which flushed buffers enter the Global Workspace. Wiggins (2012c) suggests competition between generators, quantified by a ‘volume’ metric (Equation 9.4), with one buffer from a generator being flushed on every event (since, for any given event, one generator must have the highest volume metric). Wiggins and Forth (2015), and subsequently Forth et al. (2016), suggest a mechanism more in line with perceptual models (Pearce et al., 2010b); an information content or entropy based profile maintained by each generator, which flushes when a peak or sharp rise in the profile is reached. Taken at face value, this chunking mechanism may not always flush a chunk to the Global Workspace, and may flush chunks from multiple generators at the same time. The second area relates to how predictions from higher levels in the temporal hierarchy inform predictions in the level below. If a statistical generator only models a single feature (or viewpoint) on a given temporal layer, then access to information inside generators on the layer above is necessary. Wiggins and Forth (2015) suggests that the generators are temporally aligned, and rely on counting co-occurrences between generators to build their statistical models. However, predictions of the current event rely on buffers in other generators that have yet to be flushed, and therefore, have not entered the Global Workspace, and so are not accessible to other generators. The final area is the conceptual space (Gardenfors, 2000) representations of the upper temporal layers; specifically, how they relate to multiple surface events (or chunks), and how they may be learned automatically from training data.

The aim of the following chapters is to develop and empirically test a partial, but fully functional in predictive terms, computational implementation of IDyOT. The research will focus on the first two aspects noted above; chunking mechanisms and combining predictions across temporal levels, whilst reserving the implementations of conceptual space representations, parallel parsings, and a strict Global Workspace for future research.

Chapter 10

Chunking and Prediction: A Preliminary IDyOT Implementation

10.1 Overview

This chapter aims to describe an exploratory implementation of the IDyOT architecture, focussing only on enough components to produce a functional predictive model. The hierarchical probabilistic network discussed in loose terms in §9.5 is given a concrete DBN-like implementation (§10.3.1). Different information theoretic measures related to the 'audibility' or 'volume' of generators are defined (§10.3.2) and quantitatively compared (§10.6.3). These information theoretic measures are referred to as *boundary strength measures*. A collection of techniques tasked with finding chunk boundaries by identifying relatively high points in the chunk strength measure's profile are implemented (§10.3.3), and quantitatively compared (§10.6.4). A full conceptual space implementation for representing chunks is beyond the scope of the current research, but a simpler method for labelling chunks is discussed (§10.3.4) and empirically tested (§10.6.5). Finally, a method for combining predictions on different temporal levels using multiple viewpoint techniques is presented (§10.3.1) and tested empirically (§10.6.6).

Individually, some of these components have been implemented and tested in related research. A collection of generators (§9.3.1) modelling a musical surface (or equivalent representation in another domain), capturing long and short term memory can be implemented as a multiple viewpoint system (Conklin & Witten, 1995) with the IDyOM model (Pearce, 2005). The information theoretic mechanism whereby buffers are flushed to the Global Workspace (§9.3.2), hereby referred to as the *chunking mechanism*, can be construed as a segmentation task where boundary points, corresponding to a buffer flushing to the Global Workspace, are identified at highly unexpected events or after highly uncertain contexts (Griffiths et al., 2015; Pearce et al., 2010b; Wiggins, 2012a).

A number of components discussed in the theoretical exposition of IDyOT (Chapter 9) are necessarily simplified, or not implemented in the current research. A minimalist approach is necessary to fully understand how both individual and small sets of components work and relate to one another before addressing further components. A conceptual space representation capable of labelling and positioning arbitrary sequences in a geometric space according to their information theoretic properties is non-trivial and reserved for future research. Parallel parsing with separable collections of generators (as discussed in §9.5) is handled with a simplified implementation in the form of hidden states in a DBN. The Global Workspace is implemented only implicitly. Chunks are added individually to the statistical model of a generator on the temporal level above, only after a segment boundary has been found. However, no specific definition of what information is 'inside' or 'outside' the Global Workspace is given. Finally, this initial research implements and tests only the first layer of the hierarchy, although in principle the method can be extended recursively to capture any number of layers. It is important to note that, in spite of the omissions discussed above, the preliminary IDyOT presented here is a functional predictive model, which in turn allows it to be empirically tested.

Related approaches to bottom-up chunking are discussed in §10.2, the preliminary IDyOT implementation is described in full in §10.3, a 'pen-and-paper' prediction example presented in §10.4, the motivating factors behind, and implications of, the implementation in §10.5, and various parametrisations empirically tested with mean information content in §10.6.

10.2 Related Approaches: Bottom-up Chunking and Hidden State Models

As an unsupervised learning model, IDyOT shares similarities with a number of other bottom-up approaches to sequential modelling. Here, bottom-up refers to models capable of learning higher order structure purely from statistical regularities in the surface form. Therefore, 'bottom' and 'top' are in relation to the hierarchical layers, with the bottom being the (musical) surface, and top being the higher layers in the hierarchy.[1] The following section briefly reviews and compares alternative approaches to the proposed implementation.

Probabilistic models employing hidden states, such as HMMs, provide a familiar starting point. HMMs have been widely employed in modelling symbolic music for tasks ranging from generation to classification (Allan & Williams, 2005; Chai & Vercoe, 2001; Raphael & Stoddard, 2004). An HMM assumes that observed states are statistically independent, a potential downfall in music modelling addressed by auto-regressive hidden Markov model (ARHMM) approaches, where the observed states are also conditioned on the preceding observed state (Leistikow, 2006; Rohrmeier & Graepel, 2012). HMMs and their variants can learn the hidden parameters (probability distributions) associated with their hidden states using only the observed (surface) states with the Expectation-Maximisation (EM) algorithm (Russell & Norvig, 2009, pp. 816-824). The EM algorithm uses a two-step iterative process to first provide an estimate of the sequence of observed states, and then updates the parameters of the model to maximise their likelihood. However, in order to calculate estimates efficiently the sequence must be passed both forwards and backwards, rendering the process unsuitable for an online account of musical perception (although not necessarily in an offline, memory consolidation process). Additionally, the use of explicitly hidden states reduces the impact of the approach as an explanatory model. The symbols of a hidden state do not relate in a specific manner to the observable surface and therefore do not lend themselves easily to understanding higher order cognitive structure with computational models. This observation also holds for traditional DBN models (Murphy, 2002).

The Competitive Chunker (Servan-Schreiber & Anderson, 1990) and PARSER (Perruchet & Vinter, 1998) models are two bottom-up approaches in natural language processing that form hierarchical structure in a more explicit manner. Both models have been applied to modelling melodic structure and data from behavioural studies, where the Competitive Chunker is found to outperform the PARSER model (Rohrmeier, 2010, ch. 8). The Competitive Chunker recursively builds chunks from a surface sequence until a single chunk encloses the whole sequence. For example, $ABCAB$ may be chunked as $(ABC)(AB)$, then $((AB)(C))(AB)$, and finally $(((AB)(C))(AB))$. Chunks are retrieved probabilistically according to their chunk strength, determined by a function that decays over the time since the chunk was last retrieved. Although all chunks (at first only the surface symbols) are initially given the same chunk strength, the probabilistic nature of chunk retrieval means that chunking may still be initialised from this cold starting point.

[1] This explicit definition is given to avoid confusion with the music perception and cognition literature, for example, the top-down components of Narmour's (1990) IR model are derived from exposure to music, i.e. statistical learning.

Eventually, hierarchical structure emerges as common chunks are retrieved more often, strengthening their chunk strength in a positive feedback loop. The PARSER algorithm works in a similar manner, parsing from left to right chunking in groups of one, two, or three. Chunk weights are designated by frequency occurrence, which may be reduced through *forgetting* (over time) and *interference* (through embedded structure). However, neither model captures the essence of statistical learning through modelling expectation; the motivations that underpins the current thesis (see §2.2).

The implementation presented below aims to address these issues, with probabilistic models that naturally capture expectation driving the chunking mechanism in finding higher order structure.

10.3 Model Description

Overall, this implementation of IDyOT can be viewed as a stratified DBN,[2] with a specialised upper layer, specific probabilistic rules governing dependencies between nodes, and constrained to predict only the subsequent symbol on each layer. Fundamentally, the model calculates probability distributions for the next event given past evidence from which information content and entropy of individual events can be calculated, in much the same way as a multiple viewpoint system (see Chapter 3). Events from a musical surface (see §4.2) are represented with a multiple viewpoint scheme, and therefore, much of the research presented in Part II of the thesis is applicable to this implementation. The probabilities of perceived events are calculated by the DBN using exact inference (§10.3.1), which is used to define the profile of a boundary strength measure (§10.3.2), used to indicate boundaries. A chunking mechanism (§10.3.3) selects relatively high points in the boundary strength measure's profile, marking the start of a new chunk and sending the previous chunk up to the next level (equivalent to flushing a buffer to the Global Workspace). The chunk is labelled from a finite set of symbols using an equality function (§10.3.4) that uses a viewpoint to test equality between sub-sequences of surface events. A novel unsupervised training procedure is introduced to imitate online learning from an initially empty statistical model (§10.3.5). The training procedure makes multiple passes (training epochs) over the training data to take into account the fact that statistical structure in the upper layers of the hierarchy requires statistical structure in the lower layers to be found first.

[2] Stratified in the sense that nodes are organised into layers, with edges permitted only between nodes in adjacent layers.

10.3.1 A DBN IDyOT

The DBN implementation of IDyOT is naturally described in time-slices (Murphy, 2002). A time slice, $j \in \mathbb{Z}^+$, represents a perceptual time frame in the mind of the listener, occurring as one event is perceived and before the next event begins. It may contain the previous and current events in the event sequence, indexed by i, such that $i \leq j$ for all surface events in a time slice.

A time slice consists of four nodes. The current surface event to be predicted, ${}^{j}_{k}E^{i}$, is the *query variable* (Murphy, 2002), and denoted by the tuple $\left({}^{j}E^{i},\ k\right)$.[3] The event itself, ${}^{j}E^{i}$, is the surface event of a multiple viewpoint representation: ${}^{j}E^{i} \in [\tau_{b_1}] \times ... \times [\tau_{b_n}]$ where τ_b are basic attributes (c.f. Equation 3.1). $k \in \mathbb{Z}^+$ indexes the position of the event in the current chunk, such that when $k = 1$ the event is the first of a new chunk, when $k = 2$ it is the second, etc. In practice, only predictions over the alphabet of ${}^{j}E^{i}$ are made, k merely informs the conditional probability between the current event and current chunk by indexing the location in the chunk. The surface events making up the context of ${}^{j}E^{i}$ are represented by a single *evidence variable*, ${}^{j}e_{1}^{i-1}$, a sequence $\left[e^1, e^2, ..., e^{i-1}\right]$ where each $e \in [\tau_{b_1}] \times ... \times [\tau_{b_n}] \times C$, and where C is a boolean signifying the start of a new chunk. This allows any previous chunk to be deterministically retrieved from the surface context. On the upper layer, the current chunk is the *hidden variable*, ${}^{j}_{k}V$, a tuple denoted by $\left({}^{j}V\ ,\ k\right)$. ${}^{j}V \in \mathcal{A}$, where $\mathcal{A}$ is a finite set of symbols representing the chunk alphabet (i.e. the alphabet from which the symbols representing the chunk are drawn), and $k \in \mathbb{Z}^+$ is the chunk index. Chunk symbols in $\mathcal{A}$ map onto viewpoint sequences of surface events such that each symbol in $\mathcal{A}$ maps onto a unique $\Phi_{\tau_{c_1} \otimes ... \otimes \tau_{c_n}}\left(e_x^y\right)$ where $\tau_{c_1} \otimes ... \otimes \tau_{c_n}$ is the viewpoint governing chunk equality (see §10.3.4), and x and y are the first and last event indices of a chunk. Again, in practice only probability distributions over $\mathcal{A}$ are calculated, with k serving as a chunk index. Finally, ${}^{j}u \in \mathcal{A}$ is another evidence variable representing the chunk immediately preceding ${}^{j}V$. Note that in general, seen variables are denoted by lower case letters, and unseen by upper case.

[3] Some variables in the DBN are represented as a tuple to easily separate the variable itself and the chunk index. This is notationally convenient as some of the conditional probability distributions are invariant to the chunk index, but others are not.

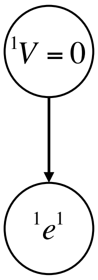


Figure 10.1: A Bayesian network representing the prior probabilities of the variables in the 2TBN. Note that the surface context, e_1^{i-1}, and the previous chunk, u, are undefined for the first time slice and therefore excluded.

After Murphy (2002), a DBN is defined by an initial Bayesian network determining the prior probability of the variables (Figure 10.1), and a two-slice temporal Bayesian net (2TBN) determining the conditional probabilities of nodes within a time slice (Figure 10.2). If $\mathbf{Z}_j$ is a vector of length X containing all random variables in a time slice j, and $parents(Z)$ a function that returns all parent nodes of a node Z, then a 2TBN may be factorised using the chain rule as follows:

$$p\left(\mathbf{Z}_j|\mathbf{Z}_{j-1}\right) = \prod_{x\in 1:X} p\left({}_xZ_j|parents({}_xZ_j)\right). \tag{10.1}$$

Parent nodes may be in the current or previous time slice, and the topology of the DBN is a directed acyclic graph.

The 2TBN of IDyOT (Figure 10.2) is characterised by four conditional probability distributions: $p({}^jE^i|{}^je_1^{i-1})$, $p({}^j_kE^i|{}^j_kV)$, $p({}^j_kV|{}^je_1^{i-1})$, and $p({}^jV|{}^ju)$. Two conditionally degenerate probability distributions: $p({}^ju|{}^je_1^{i-1})$, and $p({}^je_1^{i-1}|{}^{j-1}e_1^{i-2}, {}^{j-1}e^{i-1})$, complete the network.[4] At this point the model departs from a traditional DBN, as conditional probability distributions are estimated in a specific manner, rather than employing the EM algorithm and counting co-occurrences between variables. This is partly a result of incorporating multiple viewpoint techniques into the model, and partly because, unlike normal DBNs, the chunk alphabet, $\mathcal{A}$, consists of symbols that specifically relate to the surface layer. In a normal DBN there is no such relation as the alphabet of symbols is arbitrary without internal or relational structure.

[4] A conditionally degenerate probability distribution is a conditional distribution, such as $p(a|b)$, where a takes a single value, and is deterministic given b with a probability of 1.

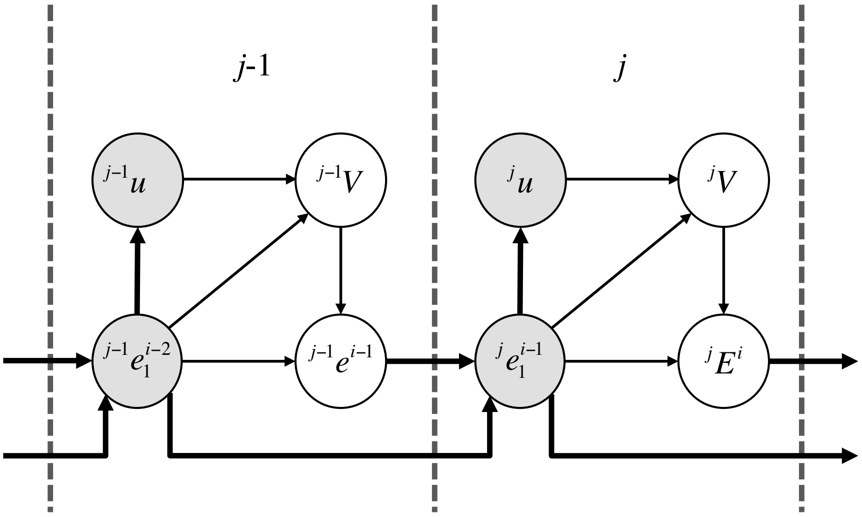


Figure 10.2: A 2TBN of the predictive components of IDyOT. Clear nodes indicate nodes that are, in general, unobserved, and grey nodes, observed. From the perspective of time slice j, lower case random variables indicate evidence variables, and upper case, hidden or query variables. Arrows indicate dependency between nodes, which may be deterministic (large arrows). Within a slice the nodes are the surface event being predicted: E^i, the context of the surface event, e_1^{i-1}, and on the upper layer the current chunk label, V, and the previous chunk, u. Events are indexed by i, and time slices by j.

10.3.1.1 Probability of Surface Event given Surface Context[5]

$p(E^i|e_1^{i-1})$ is the probability distribution of the next surface event given its (unbounded) context. As E is an event of a musical surface it is calculated with IDyOM; i.e. a multi-dimensional variable-order Markov model (Chapter 3), employing smoothing techniques (§5.2.1), merging basic attributes where appropriate (§5.5), and using techniques to improve derived viewpoint prediction (Chapter 6). Note that as chunk boundaries are not relevant for this calculation, they are not counted or matched in the statistical models estimating the surface probabilities.

[5] For readability and clarity, j is omitted for all equations and descriptions in the following sections where all variables occur in the same time slice.

10.3.1.2 Probability of Current Chunk given Previous Chunk

$p(V|u)$ is the equivalent prediction on the next layer up from the surface layer; the probability of the current (unseen) chunk given the previous chunk. It is worth noting that the previous chunk, ${}^{j}u$, is often not the chunk of the previous time slice, ${}^{j-1}V$, but is defined deterministically from the context of the musical surface by keeping track of chunk boundaries. The probability calculation is equivalent to a first-order Markov prediction using interpolated smoothing (Equation 5.2), and escape method C (Table 5-A), commonly referred to as Witten-Bell smoothing (Moffat, 1990; Witten & Bell, 1991). The chunk index, j, is not relevant to the present calculation, and therefore is omitted when matching symbols in ${}^{j}V$ for the statistical model. Recalling that the type count, $t(a_x^y)$, is the number of different symbol types occurring after the sequence a_x^y, $c(a|b)$ is simply a count of the number of occurrences of a, given the context b, ε is the empty sequence, and $\mathcal{A}'$ is the alphabet of symbols seen so far, the probability is given by Equation 10.2.[6]

$$\begin{aligned} p(V|u) = & \frac{c\,(V|u)}{\sum_{v\in\mathcal{A}} c\,(v|u) + t(u)} + \frac{t\,(u)}{\sum_{v\in\mathcal{A}} c\,(v|u) + t(u)} \times \\ & \left(\frac{c\,(V|\varepsilon)}{\sum_{v\in\mathcal{A}} c\,(v|\varepsilon) + t(\varepsilon)} + \frac{t\,(\varepsilon)}{\sum_{v\in\mathcal{A}} c\,(v|\varepsilon) + t(\varepsilon)} \times \frac{1}{|\mathcal{A}| - |\mathcal{A}'| + 1} \right) \end{aligned} \tag{10.2}$$

10.3.1.3 Probability of Current Surface Event given Current Chunk

$p({}_{k}E^{i}|{}_{k}V = v)$ is the probability distribution of the current surface event given the current chunk, usually an unseen variable. As the chunk symbols in V relate to sequences in the viewpoint governing chunk equality, $\tau_{c_1} \otimes ... \otimes \tau_{c_n}$, and the predictions are made of basic attributes of the surface event, ${}_{k}E^{i} \in [\tau_{b_1}] \times ... \times [\tau_{b_n}]$, a prediction over viewpoint elements in the upper layer is first made: $p({}_{k}T^{i}|{}_{k}V = v)$ where ${}_{k}T^{i} \in [\tau_{c_1}] \times ... \times [\tau_{c_n}]$. The distribution is estimated with a maximum likelihood calculation, counting the number of occurrences of a specific viewpoint element in T occurring at a chunk index k inside a chunk v relative to other viewpoint elements in T. The probability calculation uses interpolated smoothing (Equation 5.2) and escape method C (Table 5-A), as shown by Equation 10.3.

[6] Note that $t(\varepsilon) = |\mathcal{A}'|$. Also, note that for readability purposes, × rather than · is used to denote multiplication.

$$
\begin{aligned}
p({}_kT^i|{}_kV = v) = {} & \frac{c\left({}_kT^i|{}_kv\right)}{\sum_{{}_kt\in[\tau_{c_1}]\times...\times[\tau_{c_n}]} c\left({}_kt|{}_kv\right) + t\left({}_kv\right)} + \\
& \frac{t\left({}_kv\right)}{\sum_{{}_kt\in[\tau_{c_1}]\times...\times[\tau_{c_n}]} c\left({}_kt|{}_kv\right) + t({}_kv)} \times \\
& \frac{1}{|[\tau_{c_1}]\times ... \times[\tau_{c_n}]| - |[\tau_{c_1}]'\times ... \times[\tau_{c_n}]'| + 1}
\end{aligned}
\tag{10.3}
$$

The distribution over the chunk viewpoint, $p({}_kT^i|{}_kV = v)$, is converted to a distribution over the surface viewpoint, $p({}_kE^i|{}_kV = v)$, using the inverse function Ψ' as described in §3.4.5, with weighting techniques proposed in §6.3.

10.3.1.4 Probability of Current Chunk given Surface Context

$p({}_kV|e_1^{i-1})$ is the probability of the current chunk given the surface context. As surface events after the last chunk boundary are processed, a clearer picture of the current chunk emerges. Events in the surface context before the last chunk boundary do not inform the prediction, so e_1^{i-1} is reduced to e_{i-k+1}^{i-1}, such that only events in the current ongoing chunk are selected (see Equation 10.5). Additionally, the sequence of surface elements, $e \in [\tau_{b_1}] \times ... \times [\tau_{b_n}]$, must be converted to sequences of elements of $t \in [\tau_{c_1}] \times ... \times [\tau_{c_n}]$ in the viewpoint defining chunk equality (Equation 10.4). The probability calculation itself (Equation 10.6) is a maximum likelihood estimation using interpolated smoothing (Equation 5.2) and escape method C (Table 5-A), with the specialised count function $c_c(a|b)$ and type count $t_c(a)$ indicating that chunk boundaries are used in counting and matching sequences of e_{i-k+1}^{i-1} (in contrast to $p(E^i|e_1^{i-1})$, §10.3.1.4). To clarify, since the index $i - k + 1$ marks the start of a new chunk, only sequences of the surface context where the first event is the start of a new chunk, and every other event is not the start of a chunk (i.e. an ongoing chunk sequence), are matched.

$$
t_{i-k+1}^{i-1} = \Phi_{\tau_{c_1}\otimes...\otimes\tau_{c_n}}\left(e_{i-k+1}^{i-1}\right) \tag{10.4}
$$

$$
\begin{aligned}
p\left({}_kV|{}_ke_1^{i-1}\right) &= p\left({}_kV|e_{i-k+1}^{i-1}\right) \\
&= p\left({}_kV|t_{i-k+1}^{i-1}\right)
\end{aligned}
\tag{10.5}
$$

$$p\left({}_kV|t_{i-k+1}^{i-1}\right) = \frac{c_c\left({}_kV|t_{i-k+1}^{i-1}\right)}{\sum_{{}_kv\in\mathcal{A}} c_c\left({}_kv|t_{i-k+1}^{i-1}\right) + t_c(t_{i-k+1}^{i-1})} + \frac{t_c(t_{i-k+1}^{i-1})}{\sum_{{}_kv\in\mathcal{A}} c_c\left({}_kv|t_{i-k+1}^{i-1}\right) + t_c(t_{i-k+1}^{i-1})} \times \frac{1}{|\mathcal{A}| - |\mathcal{A}'| + 1} \tag{10.6}$$

10.3.1.5 Updating the Surface Context of the Following Time Slice

$p({}^je_1^{i-1}|{}^{j-1}e_1^{i-2}, {}^{j-1}e^{i-1})$ is deterministic, simply appending the predicted surface event of the previous time slice onto the end of the surface context of the previous time slice: ${}^{j-1}e_1^{i-2}\|{}^{j-1}e^{i-1}$ to give ${}^je_1^{i-1}$ with a probability of 1.

10.3.1.6 Finding the Previous Chunk Given the Surface Context

Finally, $p({}^ju|{}^je_1^{i-1})$ is also deterministic, extracting the previous chunk by finding the previous, and previous but one surface events that coincide with chunk boundaries. Let s be the highest index of an event with a chunk boundary, and s' be the second highest index of an event with a chunk boundary. The chunk label is assigned by converting the sequence using the viewpoint governing chunk equality with the function $\Phi_{\tau_{c_1}\otimes\ldots\otimes\tau_{c_n}}\left(e_{s'}^{s-1}\right)$ as described in §10.3.4.

10.3.1.7 Exact Inference of the Surface Event

Exact inference (Russell & Norvig, 2009, pp. 522-524) is used to estimate the prior probability of the current surface event (the query variable) given the surface context and previous chunk (the evidence variables), whilst summing out the current chunk (the hidden variable). The final line of Equation 10.7 uses ς, a function that takes any number of probability distributions over the same alphabet and combines them into a single distribution in the same manner as multiple viewpoint systems, exactly as described in §3.4.4. Probability distributions are combined using the same technique that multiple viewpoint systems use to combine viewpoint predictions, and LTM-STM predictions (§3.4.4); a geometric weighting controlled by a bias favouring probability distributions with relatively low entropy (Equation 3.7). This approach reduces the sparsity of the distributions considerably, and makes each conditional probability distribution separable.

$$\begin{aligned}
p\left({}_kE^i|{}_ke_1^{i-1},u,{}_kV\right) &= \frac{p\left({}_kE^i,{}_ke_1^{i-1},u,{}_kV\right)}{p\left({}_ke_1^{i-1},u\right)}\\
&= \frac{\sum_{{}_kv\in\mathcal{A}} p\left({}_kE^i,{}_ke_1^{i-1},u,{}_kv\right)}{p\left({}_ke_1^{i-1},u\right)}\\
&= \frac{p\left(e_1^{i-1},u\right)\sum_{{}_kv\in\mathcal{A}} p\left(E^i|e_1^{i-1},{}_kv\right)p\left({}_kv|e_1^{i-1},u\right)}{p\left({}_ke_1^{i-1},u\right)}\\
&= \sum_{{}_kv\in\mathcal{A}} p\left({}_kE^i|{}_ke_1^{i-1},{}_kv\right)p\left({}_kv|{}_ke_1^{i-1},u\right)\\
&= \sum_{{}_kv\in\mathcal{A}} \varsigma\Big(p\left({}_kE^i|{}_ke_1^{i-1}\right),p\left({}_kE^i|{}_kv\right)\Big)\varsigma\Big(p\left({}_kv|{}_ke_1^{i-1}\right),p\left({}_kv|u\right)\Big)
\end{aligned} \tag{10.7}$$

10.3.2 Boundary Strength Measures

IDyOT processes a sequence of surface events calculating the probability, $p\left({}_kE^i|{}_ke_1^{i-1},u,{}_kV\right)$, of each one as they are perceived. Following the empirical behavioural observation that segment boundaries in temporal sequences coincide with low probability, difficult to predict events (Pearce et al., 2010b; Saffran et al., 1996; Saffran et al., 1999), the probabilities are used to define a boundary strength profile, with relatively high values in the profile indicating the start of a new chunk.

Four information theoretic measures, known as *boundary strength measures*, are candidates to define the boundary strength profile. The information content, $h\left({}_ke^i|e_1^{i-1},u,{}_kV\right)$, of an event given the surface context and previous chunk is the perceived expectedness (Pearce et al., 2010c) of an event (Equation 10.8). The entropy, $H\left({}_kE^i|e_1^{i-1},u_kV\right)$, of the probability distribution over the predicted event is the perceived uncertainty (Hansen & Pearce, 2014) experienced before the event occurs (Equation 10.9). Both of these boundary strength measures have surface equivalents that ignore the upper layer: surface information content, $h_s\left(e^i|e_1^{i-1}\right)$, is given in Equation 10.10, and surface entropy, $H_s\left(E^i|e_1^{i-1}\right)$ by Equation 10.11.

$$h\big({}_ke^i|{}_ke_1^{i-1},u,{}_kV\big) = -\log_2 p\big({}_ke^i|{}_ke_1^{i-1},u,{}_kV\big) \tag{10.8}$$

$$H\left({}_kE^i|{}_ke_1^{i-1}, u, {}_kV\right) = -\sum_{e\in[\tau]} p\left(e|{}_ke_1^{i-1}, u, {}_kV\right)\log_2 p\left(e|{}_ke_1^{i-1}, u_kV\right) \tag{10.9}$$

$$h_s\big(e^i|e_1^{i-1}\big) = -\log_2 p\big(e^i|e_1^{i-1}\big) \tag{10.10}$$

$$H_s\left(E^i|e_1^{i-1}\right) = -\sum_{e\in[\tau]} p\left(e|e_1^{i-1}\right)\log_2 p\left(e|e_1^{i-1}\right) \tag{10.11}$$

Bearing in mind that summing out the hidden chunk v is required in order to calculate $p\big({}_kE^i|{}_ke_1^{i-1}, u, {}_kV\big)$ (see Equation 10.7) h, H, and h_s place chunk boundaries retrospectively; once the current event is processed they are placed before it so that it then becomes the first event of a chunk.[7] However, the surface entropy, H_s, is able to place chunk boundaries prospectively, as soon as the probability distribution of the surface event given the surface context is calculated, a chunk boundary may, potentially, be placed before the current event is perceived. This allows the chunk alphabet of the upper layer, $\mathcal{A}$, to be constrained as events of a chunk are perceived, for example, if an x is perceived any chunks that do not begin in x can be temporarily removed from $\mathcal{A}$. This is not the case for h, H, or h_s, where at any point in the sequence a chunk boundary may be retrospectively placed. This would flush the current chunk, meaning that the new current chunk is simply the empty sequence (no events have occurred since the last chunk boundary). Therefore, no symbols from $\mathcal{A}$ can be excluded, so at any point in prediction, all symbols in $\mathcal{A}$ are possible. It is hypothesised that this will afford the surface entropy, H_s, a distinct advantage as a chunk measure under empirical testing (§10.6.3).

10.3.3 Chunking Mechanisms

Given a profile of a boundary strength measure, a *chunking mechanism* is used to quantify events with relatively high boundary strengths, signifying chunk boundaries. A number of potential chunking mechanisms are defined in the current section (following a similar

[7] In the current implementation this is still true for the overall entropy, H_s. This is because $p\left(E^i|e_1^{i-1}, u, V\right)$ depends on V as it requires $p\left(v|e_1^{i-1}, u\right)$ to be calculated at the same time. If the calculated probabilities cause the overall entropy to trigger a chunk, V and u will both update, changing both probability distributions just calculated for the time slice. A perceptually and cognitively viable solution is unclear, and so the simpler approach of placing chunk boundaries retrospectively is implemented in the current research.

approach to Pearce et al., 2010b), to be empirically compared in §10.6.4. Let S^i be the boundary strength of an event at index i, and d be a threshold that triggers a new chunk. A new chunk may be triggered when S^i exceeds an absolute threshold (Equation 10.12), the delta between consecutive boundary strengths exceeds a threshold (Equation 10.13), or the ratio between consecutive boundary strengths exceeds a threshold (Equation 10.14).

$$\textit{absolute:} \quad d < S^i \tag{10.12}$$

$$\textit{delta:} \quad d < S^i - S^{i-1} \tag{10.13}$$

$$\textit{ratio:} \quad d < \frac{S^i}{S^{i-1}} \tag{10.14}$$

The three mechanisms above are fairly simplistic, failing to take into account much of the relative context. Therefore, three further mechanisms are defined using a weighted window to find the mean boundary strength of the recent context. Three types of windows are used to define the weight at an index, w^n, where L is the window length spanning from the first event to the event before the one being predicted.[8] For a uniform window $w^n = 1$, for a triangular weighted window $w^n = n$, and for an exponential decay weighted window $w^n = 0.5^{L-n}$. A chunk boundary is signalled if the current boundary strength is d standard deviations above the weighted mean boundary strength (Equation 10.17), and if $S^i > S^{i-1}$. This additional constraint, following the approach of Pearce et al. (2010b), ensures that sections of the sequence with consistently high boundary strengths do not persistently trigger chunks.

$${}_w\bar{S}_1^L = \frac{\sum_{n=1}^{L} w^n S^n}{\sum_{n=1}^{L} w^n} \tag{10.15}$$

$${}_wVar\left(S_1^L\right) = \frac{\sum_{n=1}^{L} w^n \left(S^n - {}_w\bar{S}_1^{i-1}\right)}{\sum_{n=1}^{L} w^n} \tag{10.16}$$

$$d < \frac{S^i - {}_w\bar{S}_1^L}{\sqrt{Var\left(S_1^L\right)}} \tag{10.17}$$

[8] This is the usual case, where $L = i - 1$, although shorter fixed window length may be used.

10.3.4 Labelling Chunks

Once a chunk boundary has been triggered the completed chunk is labelled and added to the relevant statistical models used to estimate $p({}_kE^i|_kV)$, $p({}_kV|e_1^{i-1})$, and $p(V|u)$. This is equivalent to the buffer of a generator in IDyOT flushing to the Global Workspace and storing the resulting chunk to memory. A *chunk equality viewpoint* is used to match equivalent chunks and assign them the same label from the finite alphabet $\mathcal{A}$. If all symbols have been assigned in $\mathcal{A}$, a special, reserved symbol is assigned to the chunk. The reserved symbol does not acquire any counts, and so will always produce a uniform distribution when predicting $p({}_kE^i|_kV)$. A chunk equality viewpoint may be a primitive, linked, or merged viewpoint, denoted by $\tau_{c_1}\otimes ... \otimes\tau_{c_n}$. The viewpoint must fully predict all of the target basic attributes of the musical surface, in much the same way that the predictive viewpoints of a multiple viewpoint system must fully predict the target basic attributes. Predicting $p({}_kE^i|_kV)$ and $p({}_kV|e_1^{i-1})$ requires that both the chunk symbol and its associated sequence of elements in the chunk equality viewpoint are stored to memory. In practice, storing the chunk viewpoint sequences is implemented as a suffix tree whose branches are the sequence of viewpoint elements of a chunk, starting at the root and ending at a leaf (see Figure 10.3). If a viewpoint element is undefined, such that $\Psi_{\tau_{c_1}\otimes ... \otimes\tau_{c_n}}\left(e_1^i\right) = \perp$, when calculating $\varsigma\Big(p\left(E^i|e_1^{i-1}\right), p\left(E^i|v\right)\Big)$ then $p({}_kE^i|_k^jV)$ is removed from the prediction combination in the same manner that undefined viewpoints are removed from viewpoint combination (see §3.4.4). To clarify, if $\Psi_{\tau_{c_1}\otimes ... \otimes\tau_{c_n}}\left(e_1^i\right) = \perp$ then $\varsigma\Big(p\left(E^i|e_1^{i-1}\right), p\left(E^i|v\right)\Big) = p\left(E^i|e_1^{i-1}\right)$.

10.3.5 Training Procedure

A new multi-pass training procedure is proposed when training and testing IDyOT, modifying the simple one-shot training procedure of IDyOM. When training IDyOM, the corpus is divided into training and test sets. The LTM+ is built by adding sequences (unbounded context plus the current event) to the PPM suffix tree on an event by event basis. The STM is not built during the training phase, but built during the test phase, and emptied at the end of each piece. Sequences are also added to the LTM+ during the test phase. This process is repeated k times if k-fold cross validation is used, with the performance of the model reported as the mean information content over all k test sets.

This training procedure is adjusted for IDyOT to more closely mimic a human learning paradigm starting form zero knowledge, and to account for the statistical structure of the upper layer. As before, the corpus is divided into training and test sets, with an LTM+ and an STM respectively modelling structure within the corpus as a whole, and within individual pieces. However, only the surface layer uses the LTM+ and STM, essentially behaving as an IDyOM model. Counts in the upper layers are retained in a separate LTM+, with no STM. The training phase proceeds over a number of epochs, each epoch is a complete pass (and indeed parse) of the training data. At the start of the first epoch the LTM+ of both the surface and upper layers are empty, with sequences added on an event by event basis to both of the LTM+ and the STM. However, unlike IDyOM, predictions in the form of probability distributions of the following event are also made, identifying chunks and influencing predictions from the top layer exactly as described above (§10.3.1 - §10.3.4). The multiple epoch approach is necessary because, unlike IDyOM, the statistical structure of the upper layer is defined purely by the predictions of the surface layer. Initially chunk boundaries may be somewhat chaotic before the surface layer model of IDyOT has had enough time to identify statistical patterns. However, as the training procedure progresses the chunk boundaries and the statistical structure of the upper layer should become more consistent (assuming that they are learning structure).

The maximum number of training epochs can be predefined, or a stopping criteria used to identify when the chunk boundaries have settled. One stopping criteria uses Cohen's Kappa, κ, to measure the agreement in chunk boundaries over all events between two successive epochs after accounting for chance agreement (Equation 10.18). Below, c is a boolean set to T if an event is a chunk boundary, otherwise F. The number of corresponding events in the k^{th} and $(k-1)^{th}$ epochs with values of c_k and c_{k-1} respectively is given by $n_{c_k c_{k-1}}$, a + signifying a sum over both T and F values for c for the epoch.[9] The stopping criteria is when κ reaches a threshold, for example when $\kappa > 0.95$.

$$\begin{aligned} p_e &= \frac{1}{n_{++}} \sum_{c \in \{T,F\}} n_{c+} \cdot n_{+c} \\ p_o &= \frac{n_{TT} + n_{FF}}{n_{++}} \\ \kappa &= \frac{p_o - p_e}{1 - p_e} \end{aligned} \tag{10.18}$$

A single test parse over the testing set is carried out in a similar manner to IDyOM, the LTM+ of the surface and upper layers retains the statistical structure from the training phase, whilst the surface STM is emptied for each piece. Mean information

[9] For example, n_{TF}, is the number of events that have a chunk boundary in epoch k, but not in epoch $k-1$, and n_{T+} is the total number of chunk boundaries in epoch k.

content and other performance statistics are calculated from the test phase alone. Like IDyOM, the process may be repeated k times for a k-fold validation.

10.4 Prediction Illustration

An illustrative example of IDyOT predicting a single event is presented to supplement the formal exposition of the implementation given in §10.3. The basic attribute, τ_b, predicted is `Letter`, which has a small domain of five symbols: $[\texttt{Letter}] = \{\texttt{A},\texttt{B},\texttt{C},\texttt{D},\texttt{E}\}$. `Vowel` is a derived viewpoint, τ_c, used as the chunk equality viewpoint (§10.3.4), simply indicating with a boolean if a `Letter` element is a vowel: $[\texttt{Vowel}] = \{\texttt{T},\texttt{F}\}$. The model has processed the following sequence of letters, with commas indicating chunk boundaries.

`BED, DEAD, BED, DEAD, BED, DEAD, BEAD, BEC, DEAD, BE` ...

The prediction task is to predict the next `Letter` symbol. The 'pen-and-paper' example calculates probabilities as described in §10.3.1, with a few simplifications aimed at making the statistical process clearer for the purposes of this illustration only. No smoothing is used for any of the four probability distributions, and the surface layer is bounded to a straightforward first-order Markov model. Explicitly:

$$p(E^i|e_1^{i-1}) = \frac{c(E^i|e^{i-1})}{\sum_{e\in[\texttt{Letter}]} c(e|e^{i-1})} \tag{10.19}$$

$$p({}_kT^i|{}_kv) = \frac{c({}_kT^i|{}_kv)}{\sum_{t\in[\texttt{Vowel}]} c(t|{}_kv)} \tag{10.20}$$

$$p(V|u) = \frac{c(V|u)}{\sum_{v\in\mathcal{A}} c(v|u)} \tag{10.21}$$

$$p({}_kV|t_{i-k+1}^{i-1}) = \frac{c({}_kV|t_{i-k+1}^{i-1})}{\sum_{{}_kv\in\mathcal{A}} c({}_kv|t_{i-k+1}^{i-1})}. \tag{10.22}$$

Note that in Equation 10.20, $p({}_kT^i|{}_kv)$ is converted to $p({}_kE^i|{}_kv)$ using $\Psi'_{\texttt{Vowel}}(t)$ (see §3.4.5), and in Equation 10.22, $t_{i-k+1}^{i-1} = \Phi_{\texttt{Vowel}}\left(e_{i-k+1}^{i-1}\right)$. Note also that counts may span chunk boundaries for the surface layer Markov model. The chunk alphabet consists of only two symbols, $\mathcal{A} = \{\mathbf{X}, \mathbf{Y}\}$; the chunk counts for the sequence are given in

Table 10-A and the associated chunk sequences stored as a prefix tree (Figure 10.3).

Table 10-A: Frequency counts of chunks used in a 'pen-and-paper' example.

$\Phi_{\texttt{Letter}}$	$\Phi_{\texttt{Vowel}}$	Symbol	Count
BEC	TFT	**X**	1
BED	TFT	**X**	3
BEAD	TFFT	**Y**	1
DEAD	TFFT	**Y**	4

Figure 10.4 shows the state of the DBN the moment before predicting the next symbol. The surface context of the current time slice is all 34 surface events processed so far, and the previous symbol `Y` relates to the chunk `DEAD`. The prediction is taking place mid-chunk, so far only `BE` has been processed in the current chunk. By observing the sequence of surface events processed, and the frequency counts of the chunks (Table 10-A) one can build an intuitive picture of the probabilistic predictions taking place. The surface symbol `E` is commonly followed by both `A` and `D`, so will be relatively uncertain between those two symbols. On the upper layer, however, `BED` often follows `DEAD`, which implies a stronger preference for `D` as the next symbol on the surface layer.

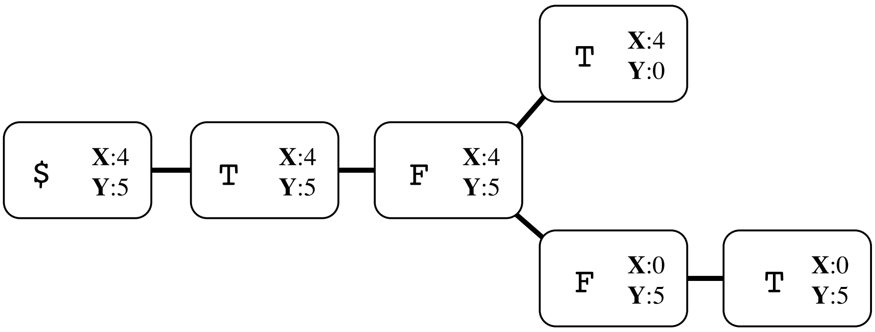


Figure 10.3: Prefix tree of chunk sequences in a 'pen-and-paper' example.

Table 10-B shows the individual probability distributions, $p(E^i|e^{i-1})$ and $p({}_kT|{}_kV = v)$, of the surface event given the surface context, and the surface event given the current chunk. Note that the surface event given the current chunk must first be predicted through the chunk equality viewpoint, `Vowel`, before being converted to a distribution over [`Letter`] with $\Psi'_{\texttt{Vowel}}$. These probability distributions are combined (Table 10-C) with a weighted geometric combination, $\varsigma(p_x, p_y)$, as described in §3.4.4. Overall, Table 10-C shows that if the current chunk is `X`, a surface symbol of `D` is more likely than `C` with no other symbols possible, whilst if the current chunk is `Y` only a surface symbol of `A` is possible. In effect, the current chunk (the hidden variable V) serves to both constrain and inform the distribution of the surface event.

Table 10-B: Individual probability distributions associated with predicting the current surface event, E^i, of a ‘pen-and-paper’ example.

$[\tau_b]$	$p(E^i \vert e^{i-1})$	$[\tau_c]$	$p({}_kT \vert {}_kV = \mathbf{X})$	$p({}_kT \vert {}_kV = \mathbf{Y})$	$[\tau_b]$	$p({}_kE^i \vert {}_kv = \mathbf{X})$	$p({}_kE^i \vert {}_kV = \mathbf{Y})$
A	$\frac{5}{9}$				A	$\frac{0}{3}$	$\frac{1}{2}$
B	$\frac{0}{9}$	F	$\frac{4}{4}$	$\frac{0}{5}$	B	$\frac{1}{3}$	$\frac{0}{2}$
C	$\frac{1}{9}$				C	$\frac{1}{3}$	$\frac{0}{2}$
D	$\frac{3}{9}$	T	$\frac{0}{4}$	$\frac{5}{5}$	D	$\frac{1}{3}$	$\frac{0}{2}$
E	$\frac{0}{9}$				E	$\frac{0}{3}$	$\frac{1}{2}$

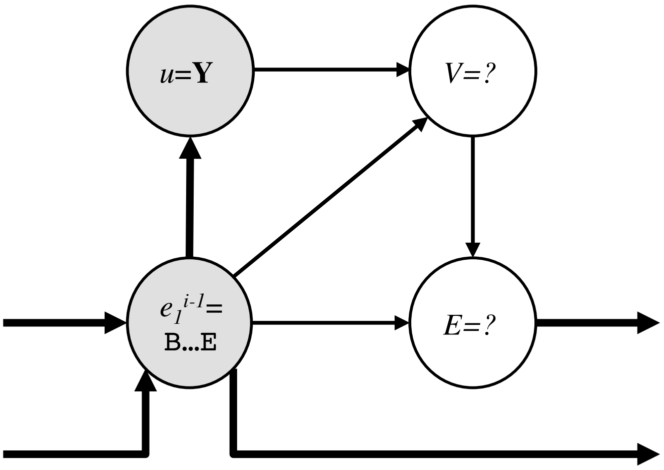


Figure 10.4: Current state of the IDyOT DBN at the point of prediction in a 'pen-and-paper' example.

As the current chunk, V, is a hidden variable it must be summed out. The predictions associated with the current chunk are given in Table 10-D, showing both the conditioning of the previous chunk, u, and the surface context, e_{i-k+1}^{i-1}, preferring X over Y. The summing out itself is completed in Table 10-E, with the final probability distribution in the final column showing a D is the most likely consequent symbol, with A and C given lower probabilities, and B and E given a probability of 0 as a result of the constraints from the current chunk on the upper layer.[10]

Table 10-C: Combined probability distributions associated with predicting the current surface event, E^i, of a 'pen-and-paper' example.

$[\tau_b]$	$p({}_kE^i \vert e^{i-1}, {}_kV = \mathbf{X}) =$ $\varsigma\Big(p(E^i \vert e^{i-1}), p({}_kE^i \vert {}_kV = \mathbf{X})\Big)$	$p({}_kE^i \vert e^{i-1}, {}_kV = \mathbf{Y}) =$ $\varsigma\Big(p(E^i \vert e^{i-1}), p({}_kE^i \vert {}_kV = \mathbf{Y})\Big)$
A	0.000	1.000
B	0.000	0.000
C	0.356	0.000
D	0.644	0.000
E	0.000	0.000

[10] If smoothing is used all symbols in the probability distribution will have a non-negative probability.

Table 10-D: Individual and combined probability distributions associated with predicting the current chunk, V, of a 'pen-and-paper' example.

$\mathcal{A}$	$p(V\|u)$	$p(V\|e_s^{i-1})$	$p(V\|e_s^{i-1},u) = \varsigma\Big(p(V\|u), p(V\|e_{i-k+1}^{i-1})\Big)$
X	$\frac{2}{3}$	$\frac{3}{4}$	0.713
Y	$\frac{1}{3}$	$\frac{1}{4}$	0.287

Table 10-E: Individual and combined probability distributions associated with predicting the current surface event, E^i, given the surface context and upper layer chunks in a 'pen-and-paper' example.

$[\tau_b]$	$p({}_kE^i\|e^{i-1}, {}_kv = \mathbf{X})\times$ $p({}_kv = \mathbf{X}\|e^{i-1}_{i-k+1}, u)$	$p({}_kE^i\|e^{i-1}, {}_kv = \mathbf{Y})\times$ $p({}_kv = \mathbf{Y}\|e^{i-1}_{i-k+1}, u)$	$p({}_kE^i\|e^{i-1}, u, {}_kV) =$ $\sum_{{}_kv\in\mathcal{A}} p({}_kE^i\|e^{i-1}, {}_kv)\times$ $p({}_kv\|e^{i-1}_{i-k+1}u)$
A	0.000	0.287	0.287
B	0.000	0.000	0.000
C	0.254	0.000	0.254
D	0.459	0.000	0.459
E	0.000	0.000	0.000

10.5 Implementation Motivations and Implications

Certain aspects of the IDyOT implementation presented in §10.3 have various cognitive implications, to be considered in the current section. With these in mind, the motivations behind the precise implementation presented are discussed.

10.5.1 Perceptual and Cognitive Motivations

The underlying motivations behind IDyOT are that expectation driven statistical learning is able to account for the unsupervised learning of complex sequential structure. Where possible, the implementation uses cognitively and perceptually validated processes. Specifically, the prediction of the subsequent surface events is essentially carried out by IDyOM, a computational model capable of accounting for a range of perceptual and cognitive behaviours including expectation (Pearce et al., 2010c; Pearce & Wiggins, 2006), uncertainty (Hansen & Pearce, 2014), and memory (Agres et al., 2017).[11] Furthermore, the chunking mechanism (§10.3.3) that triggers chunk boundaries with low probability events has empirical foundations in behavioural studies with both pitch

[11] §2.4.3 provides a more detailed review.

(Pearce et al., 2010b; Saffran et al., 1999), and speech (Saffran et al., 1996) sequences.

10.5.2 The Global Workspace

The Global Workspace as envisaged by Baars (1988) and consequently by Wiggins and Forth (2015) is not explicitly implemented in the preliminary IDyOT model proposed in §10.3. The main motivation behind this decision is to enable the combination of predictions from different generators; in the current implementation these are viewpoint predictions, LTM predictions, STM predictions, and upper layer predictions. As identified in §9.7, generators in a strict Global Workspace architecture may not combine their predictions unless they enter the Global Workspace, which only occurs when they flush their buffers. Nevertheless, the essence of the Global Workspace exists in the implementation; generators flush buffers to the workspace, which are subsequently consolidated as a chunk and added to a statistical model.

10.5.3 Parallel Parsing

Similarly, although a mechanism for parallel parsing is not explicitly implemented, the proposed exploratory model is able to capture some notion of ambiguity in its higher order representations. The current chunk is a hidden variable, and so simultaneously holds at any given time all symbols from the (potentially constrained) chunk alphabet, $\mathcal{A}$, with a probability $p({}_kV = v|{}_ke_1^{i-1}, u)$. In calculating the overall probability of the next event given the context and previous chunk (Equation 10.7), when summing out ${}_kV$, $p({}_kV = v|{}_ke_1^{i-1}, u)$ effectively acts as an arithmetic weighting for each surface prediction, $p({}_kE^i|{}_ke_1^{i-1}, {}_kV = v)$. Another way of viewing this process is that each potential realisation of the hidden variable is itself a parallel path, where $p({}_kV = v|{}_ke_1^{i-1}, u)$ is the likelihood of the path given the preceding events and chunk.

10.5.4 Potential to Model Non-Local Dependencies

One of the key motivations behind implementing a statistically driven hierarchical model is to determine the extent to which higher order structure might be accounted for by strictly bottom-up processes (c.f. Rohrmeier, 2011). The exploratory implementation described above (§10.3) accounts for only a single hierarchical layer, so constructs such as unbounded dependencies and embedded recursive structures are not accounted for. However, in theory non-local structure may potentially be captured, bearing in mind that the current chunk, the hidden state V, may refer to a sequence of surface symbols of arbitrary length. Probabilistic influence may, therefore, pass from the surface context, e_1^{i-1}, and previous chunk, u, to any arbitrary surface event in the future. The implementation may be extended recursively upwards to any number of hierarchical layers representing increasingly abstract chunk representations, and increasingly longer time spans. However, bearing in mind that musical expectancies decay after 10-12 seconds (Farbood, 2010; Woolhouse et al., 2016), and that finite-state grammars may approximate context-free grammars with bounded recursion, a small number of layers may be sufficient to model the fundamental cognitive and perceptual processes associated with musical structure.

10.5.5 Multiple Viewpoint Systems

A final motivating factor in the initial implementation of IDyOT is to retain, where possible, the multiple viewpoint framework. The addition of the upper layer can be viewed as a further viewpoint, albeit one defined by the statistical structure of the musical surface, rather than purely the representational structure. The technique used to combine viewpoint predictions (§3.4.4) is also applied to combining predictions across layers, making for a compact model in terms of the number of different types of processes required. The implementation holds some similarities with predicting multiple melodic lines (Whorley, 2013), in that the temporal layers are separable in a manner that attributes on a single layer are not. However, a key difference between the approaches is that whilst Whorley (2013) finds statistical structure with co-occurrences between layers (with intra-linked viewpoints), which may constrain other layers (Whorley et al., 2013a), the current research explicitly makes predictions between layers.

10.6 Testing Parametrisations of IDyOT

The implementation of IDyOT presented in §10.3 contains a number of free parameters and component variations. Most of these define the methods used to signify chunk boundaries: the variety of measures used to represent the boundary strength, S, (§10.3.2), the mechanism used to identify rises in the boundary strength measure's profile (§10.3.3), as well as the chunking threshold, d, itself. Furthermore, the representation scheme of the chunk layer is governed by a chunk equality viewpoint (§10.3.4), which may be any viewpoint that fully predicts the surface layer. The chunk alphabet, $\mathcal{A}$, is finite, with a parameter controlling its size. Finally, two bias parameters control the geometric combination of probability distributions predicting the surface event, and the current chunk (§10.3.1.7), equivalent to the LTM-STM and viewpoint combination biases in a multiple viewpoint system (§3.4.4). The bias mediating the surface event prediction, $\varsigma(p(E^i|e_1^{i-1}), p({}_kE^i|{}_kv))$, will be referred to as the *event bias*, and the bias mediating the current chunk prediction, $\varsigma(p({}_kv|{}_ke_1^{i-1}), p(v|u))$, the *chunk bias.* In the following, both numerical parameters and categorical parameters (e.g. the chunking mechanism) are referred to as free parameters. The purpose of the following section is to empirically compare these parametrisations, aiming to guide future IDyOT implementations, and to begin to understand at a functional level some of the statistically-driven cognitive processes required to represent higher order structure.

10.6.1 Experimental Design

Mean information content, $\bar{h}$, (Equation 10.23) is used a heuristic to compare and quantify the performance of various IDyOT parametrisations. As before (§3.4.2), $\bar{h}$ represents the average number of bits required to encode each event, or can be viewed as the divergence between the probability distribution of the model, and the (inaccessible) stochastic process generating the training data.[12] Importantly, $\bar{h}$ serves as a valid performance heuristic between models tested over the same testing data, with events drawn from a common, finite alphabet. A 10-fold cross-validation is used to calculate $\bar{h}$.

$$\bar{h}(e_1^N) = -\frac{1}{N}\sum_{i=1}^{N} \log_2 p({}_ke^i \mid e_1^{i-1}, u, {}_kV^i) \tag{10.23}$$

IDyOT is tested by predicting chord sequences from the original *Real Book Vol. 1* (Leonard, 2012); the primary domain and corpus of the present research (Table 4-E, dataset 1). Following the findings of Chapter 5, the merged attribute `Root`$\otimes$`ChordType` is predicted, with `PosInBar` as a given attribute (see §4.5).

The complete parameter space (Table 10-F) of the collection of free parameters outlined above is too large to search exhaustively. An informal search strategy is employed, following the approach of Pearce and Wiggins (2004) in finding optimal smoothing parametrisations for IDyOM. A subset of the parameter space is tested at each stage in the search, with the parameters of the best performing model being carried forward to the next stage, repeated until a locally optimal model is produced after all subsets have been explored. Unfortunately, due to long run times and the large parameter space, this search strategy is the only practical approach to systematically exploring the parameter space. For v predictive viewpoints, each with a domain size of $|[\tau]|$, using a chunk alphabet of size $|\mathcal{A}|$, predicting on a sequence of length m, the time complexity is

[12] Also referred to as cross entropy; see Manning and Schütze (1999, pp. 74-76).

$O(|\mathcal{A}| \cdot v \cdot |[\tau]| \cdot m^2)$ (see §3.6). Despite its informal nature, this search strategy provides a useful framework to compare specific components of the model (e.g. boundary strength measures, chunking mechanisms, chunk equality viewpoints), as all other parameters are kept fixed at each stage, whilst the single parameter of interest is varied. In order to make consistent comparisons between models, the maximum number of training epochs is fixed at 2.

Table 10-F: Complete parameter space for IDyOT predicting chord sequences.

Parameter	Set of values
Boundary strength measure	$S \in \{h, H, h_s, H_s\}$
Chunking threshold	$d \in \mathbb{R}$
Chunking mechanism	$m \in \{absolute, delta, ratio, uniform\ window,$ $triangular\ window, exponential\ window\}$
Chunk alphabet size	$\lvert\mathcal{A}\rvert \in \mathbb{Z}^*$
Chunk equality viewpoint	$\tau_{c_1} \otimes \tau_{c_2} \in \{$`Root, RootInt, ChromaDist, MeeusInt`$\}\otimes$ $\{$`ChordType, FunctionType, MajType, Type`$\}$
Event bias	$b \in \mathbb{Z}^*$
Chunk bias	$b \in \mathbb{Z}^*$

The surface level predictions, $p(E^i|e_1^{i-1})$, are calculated with an IDyOM model. The best performing smoothing techniques and combination biases from §5.6 and §5.7 are retained. The only exception is that an order bound of 1 is placed on both the LTM+ and the STM. It is anticipated that the upper layer of IDyOT may, in part, subsume the variable order components of the lower layer, and so in order to distinguish between the two effects, one must be removed. In summary, an STMC1IUM-LTM+C1IM model[13] using bias weights of $b = 2$ and $b = 1$ for LTM-STM and viewpoint combination predicts the surface layer. The predictive viewpoints are the first three viewpoints selected in Figure 6.2: `Root`⊗`ChordType`⊗`PosInBar`, `RootInt`⊗`ChordType`, and `RootIntFiP`⊗`ChordType`⊗`PosInBar`. In theory, more viewpoints could be used to predict the surface layer, but as shown in all viewpoint selection runs in the current research (§5.7, §6.4.2, Chapter 7), viewpoints added later in the selection run contribute only small gains in predictive performance, whilst still adding considerably to time and memory.

Ultimately, the performance of IDyOT as a model of cognition should be assessed in its ability to correlate with human behaviour. However, such an evaluation is reserved for future research after the computational implementation has been fully developed and evaluated. Mean information content is used as a heuristic for model selection, and although in general corresponds with good correlations with human behaviour (Pearce & Wiggins, 2006), is not guaranteed to produce an optimal cognitive model. Although it is

[13] See §5.2.1.4 for details on the shorthand model notation.

only a preliminary performance metric in this sense, a rough baseline of performance for $\bar{h}$ of 3.298 bits/symbol can be found by calculating event probabilities with IDyOM using the experimental design as described above (see §10.6.7 for a more in-depth comparison between IDyOM and IDyOT).

The informal search optimising subsets of parameters roughly in order of information flow around the system proceeds as follows. After some preliminary runs (§10.6.2), boundary strength measures are compared (§10.6.3), followed by chunking mechanisms (§10.6.4), chunk equality viewpoints (§10.6.5), and event and chunk biases (§10.6.6).

10.6.2 Preliminary

A few preliminary runs of IDyOT establish a starting point in the search, and identify any unforeseen problems with the model. These initial runs were conducted on a single fold of the 10-fold cross validation, judging model performance with mean information content, the extent to which the most common chunks found match musicological expectations, and the quality of the segmentations on a few well known jazz standards in the test set.[14]

The overall information content, $h({}_ke^i|{}_ke_1^{i-1}, u, {}_kV)$, is chosen as the initial boundary strength measure (following the previous research of Pearce et al., 2010b; Wiggins, 2012a), with the *delta* chunking mechanism (Equation 10.13) proving to be simple but effective. A chunking threshold of $d = 2$ produced both a musicologically meaningful distribution of chunk types, and provided plausible segmentations of the selected jazz standards. A relative viewpoint, `RootInt`$\otimes$`ChordType`, is chosen as the chunk equality viewpoint, with this level of abstraction enabling a larger number of unique chunk sequences, referred to as chunk types, to be stored in the finite chunk alphabet. The full initial parametrisation, serving as the initial state of the informal search is summarised in Table 10-G, returning a mean information content of 4.103 bits/event with a full 10-fold cross validation.

During preliminary testing, a substantial problem associated with the finite chunk alphabet was identified, showing an alphabet size of 1,000 to be woefully insufficient to label the large number of chunk types found during the training and testing phases.[15] On average, within a fold each training epoch contains 576.8 unique unlabelled chunks (chunk types), or 630.95 non-unique unlabelled chunks (chunk tokens). To measure the

[14] An assessment of the distribution of chunk types and meaningful segmentations of individual jazz standards are explored in detail in §10.6.7 and §11.5 respectively.

[15] It is worth clarifying that the chunk alphabet is not shared across the 10-fold cross validation, but a unique mapping between symbols in the chunk alphabet and chunk sequences is established for each fold.

Table 10-G: Preliminary IDyOT parametrisation and performance metrics.

Parameter	Set of values
Boundary strength measure	$S \in \{h, H, h_s, H_s\}$
Chunking threshold	$d \in \mathbb{R}$
Chunking mechanism	*delta*
Chunk alphabet size	$1,000$
Chunk equality viewpoint	`RootInt`$\otimes$`ChordType`
Event bias	1
Chunk bias	1
Mean information content:	4.103
Chunk coverage:	23.93%

resulting impact on the test sets, a *chunk coverage* measure is defined as the percentage, by event, that the test set is covered by a labelled chunk. The low chunk coverage of 23.93% suggests that the chunk alphabet poorly represents the test data, motivating a modification to the chunk labelling method.

A closer observation of the distribution of chunk types ranked by frequency of chunks in IDyOT's memory (Figure 10.5) reveals a Zipf-like (Zipf, 1935, 1949) distribution. In a Zipf distribution the frequency of occurrences of an item is inversely proportional to its frequency rank, originating in linguistics, and later found to hold for various musical domains (Rohrmeier & Cross, 2008; Zanette, 2006). The peculiarities of the training procedure distort the distribution somewhat; a longer tail of single occurring chunk types is curtailed when the finite chunk alphabet runs out of available symbols, and a disproportionately large number of chunk types occur precisely 18 times because this is the number of times a piece is seen in a training set with the 10-fold cross validation over two training epochs. Nevertheless, the salient feature of the distribution is that the most highly ranked of the 1,620 chunk types stored in memory account for high proportions of the total number of chunk tokens stored: the top 100 ranked chunk types account for 59.5% of all chunks stored, the top 10 account for 35.4% of chunks stored, and the highest ranked chunk, the chunk sequence $[(\perp, min7), (5, 7)]$, accounts for 14.7% of chunks stored.

Given that a small number of chunk types account for a large proportion of the total number of chunk tokens, and a large proportion of the chunk alphabet is wasted with chunk types that only occur a small number of times, the following approach is taken. At the end of each training epoch all chunk types in the alphabet that have only occurred once are forgotten and removed from all statistical models, with the associated symbol in the chunk alphabet free to be reassigned in the next training epoch. Furthermore, by using this method the chunk alphabet size can be reduced to 100, with the corresponding

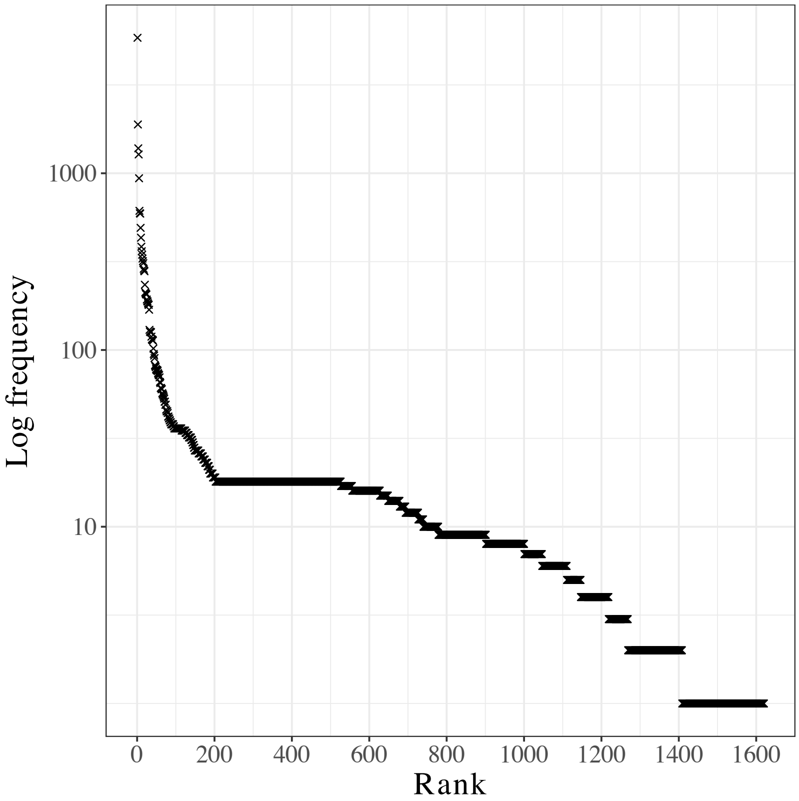


Figure 10.5: Log frequency of chunks stored by IDyOT by rank for a complete training and test procedure summed over all folds of a 10-fold cross validation.

model producing a lower mean information content of 4.071 bits/symbol, with a chunk coverage of 49.1%.

10.6.3 Testing Boundary Strength Measures

The boundary strength measure is an information theoretic measure used to create an event by event profile, the high points of which indicate chunk boundaries (§10.3.2). Four boundary strength measures are available to IDyOT: information content, $h({}_ke^i|{}_ke_1^{i-1}, u, {}_kV)$, entropy, $H({}_kE^i|{}_ke_1^{i-1}, u, {}_kV)$, surface information content $h_s(e^i|e_1^{i-1})$, and surface entropy, $H_s(E^i|e_1^{i-1})$. Surface information content has been used as a boundary strength measure in both music (Pearce et al., 2010b) and language (Griffiths et al., 2015; Wiggins, 2012a) segmentation tasks, however, it has only been evaluated in terms of correlating segmentations with expert annotated ground truths, or human behavioural data. By contrast, the current experiment provides an opportunity to evaluate all four boundary strength measures in terms of their ability to predict testing data in an information theoretically compact manner, or more simply, minimise mean information content. The parametrisations tested in the following experiment are summarised by Table 10-H. The chunking threshold, d, is not equivalent between chunk strength measures,[16] so d must be varied for each boundary strength measure.

Table 10-H: Parametrisations when testing boundary strength measures.

Parameter	Set of values
Boundary strength measure	$S \in \{h, H, h_s, H_s\}$
Chunking threshold	$0.0 \leq d \leq 8.0$
Chunking mechanism	*delta*
Chunk alphabet size	100
Chunk equality viewpoint	`RootInt`$\otimes$`ChordType`
Event bias	1
Chunk bias	1

10.6.3.1 Hypotheses

All boundary strength measures are expected to be able to segment sequences into reasonable chunks; this has already been observed in the case of information content (Griffiths et al., 2015; Pearce et al., 2010b; Wiggins, 2012a). The expectation is that meaningful segmentations provide the basis for more efficient predictions in IDyOT. Meaningful segmentations should find a balance between over segmenting (where signalling a chunk boundary at every event creates an upper layer almost equivalent to the surface) and under segmenting (where very few chunk boundaries are found, and the resulting long chunks create a very sparse statistical model). As such, when varying d an expected behaviour of the system would be to find an optimal point between these two extremes. Loosely considering various tonal harmonic parsing systems (Lerdahl & Jackendoff, 1983; Marsden, 2010; Pachet, 2000; Rohrmeier, 2011; Steedman, 1984; Ulrich, 1977) a mean chunk length of between two and five events is expected to be optimal. The surface entropy as a boundary strength measure may potentially outperform its counterparts because it is able to predict chunk boundaries prospectively, as opposed to retrospectively (as discussed in §10.3.2).

[16] Typically, information content values are higher than entropy values for the data and alphabets used in the present research.

10.6.3.2 Results

The performances of each boundary strength measure across a pre-selected set of chunking thresholds are summarised in Figure 10.6, and shown in more detail in Tables E-1, E-2, E-3, and E-4 (Appendix E). A direct comparison between boundary strength measures shows surface entropy, H_s, to return the best performance in terms of mean information content, $\bar{h}$. The lowest $\bar{h}$ of 3.706 bits/symbol for boundary strength measure H_s is found when $d = 0.0$. This significantly outperforms the best performing model ($\bar{h} = 4.063$) using any other boundary strength measure (h_s, when $d = 5.0$), as judged with a paired, one-sided t-test over pieces ($df = 347, t = 23.287, p < 0.001$, Cohen's $d = 0.371$).

Otherwise, the results are surprising, and signify highly unexpected behaviour in the system. For both information content, h, and surface information content, h_s, varying the chunking threshold, d, has little impact on overall performance; no clear optimal point can be found for either chunk measure. For h, the best performing model (when $d = 4.0$) does not statistically significantly ($df = 347, t = -0.481, p = 0.685$, Cohen's $d = -0.007$) outperform the worst performing model (when $d = 1.0$). Likewise, for h_s, the best performing (when $d = 5.0$) model fails to statistically significantly ($df = 347, t = 0.598, p = 0.275$, Cohen's $d = 0.008$) outperform the worst performing model (when $d = 0.5$). Interestingly, there is a strong trend in H_s to prefer lower chunk thresholds (resulting in shorter chunks), but the trend is reversed for H which prefers higher chunk thresholds, resulting in longer chunks. It is worth noting that for H when $d = 4.0$, the mean chunk length is 30.213 (Table E-2), which approaches the mean piece length for the dataset (43.670). By contrast, for the optimal model using H_s, when $d = 0.0$ the mean chunk length is 2.139 (Table E-4), around the lower bound what might be expected.

To summarise, surface entropy, H_s, was the best performing boundary strength measure, and therefore is retained for the following empirical comparison of chunking mechanisms. However, the lack of expected optimal chunking thresholds implies that this implementation of IDyOT does not behave as anticipated in terms of minimising mean information content with meaningful upper layer segmentations. Further parametrisations are explored in the following sections to ascertain whether this behaviour is a result of non-optimal parameters in other components of the system, or a more fundamental issue with the implementation.

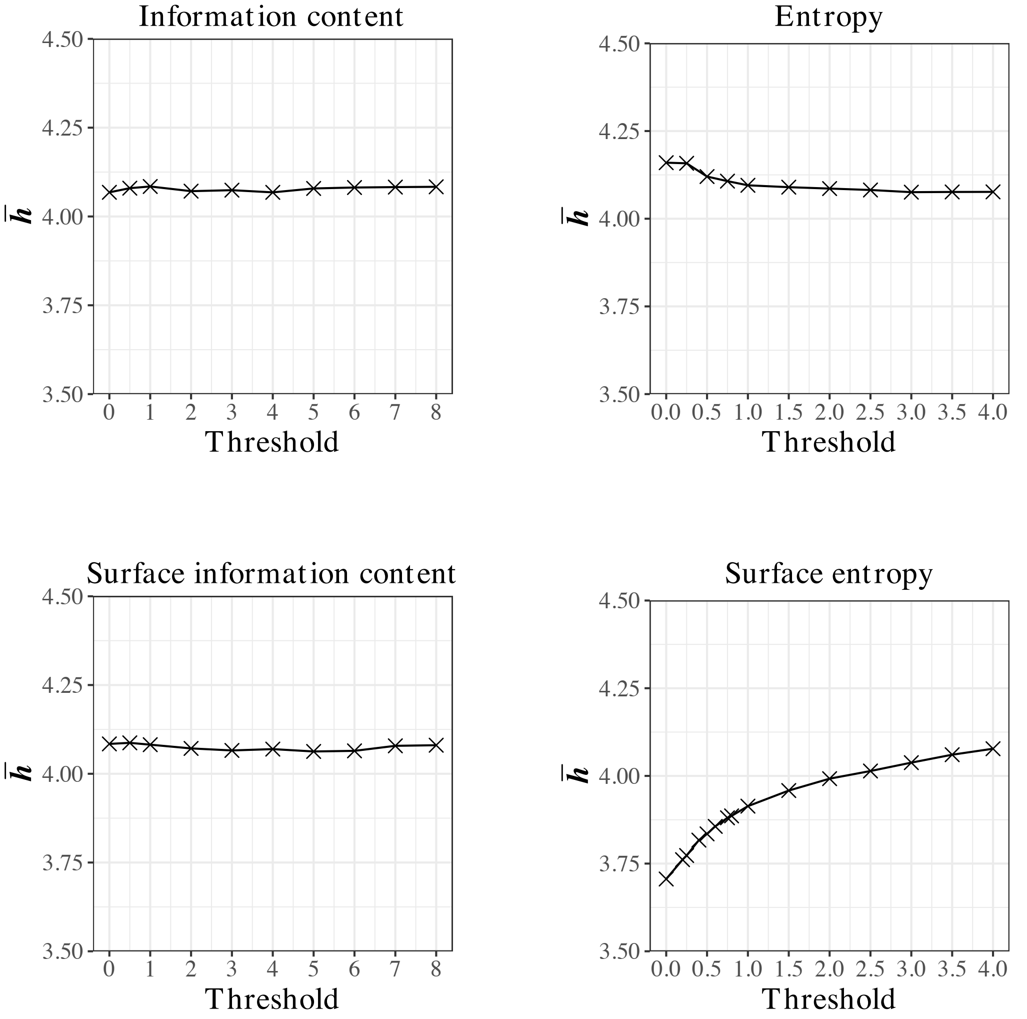


Figure 10.6: Performance of IDyOT over four boundary strength measures: information content, $h({}_ke^i|{}_ke_1^{i-1}, u, {}_kV)$, (top left), entropy, $H({}_kE^i|{}_ke_1^{i-1}, u, {}_kV)$, (top right), surface information content $h_s(e^i|e_1^{i-1})$, (bottom left), and surface entropy, $H_s(E^i|e_1^{i-1})$, (bottom right).

10.6.4 Testing Chunking Mechanisms

The chunking mechanism is the method used to identify high points in the profile of the boundary strength measure (§10.3.3). The boundary strength measure results in §10.6.3 use a relatively simple chunking mechanism: the absolute difference between successive boundary strengths, or *delta*. This chunking mechanism is potentially too simplistic as it is insensitive to the recent context of the boundary strength measure profile, or whether the boundary strength itself is relatively high or low. Pearce et al. (2010b) presents and empirically tests a more sophisticated chunking mechanism, identifying boundaries when the information content is d standard deviations above the weighted mean, calculated with a triangular window (see Equation 10.17). The following section empirically compares six chunking mechanisms: *absolute*, *delta*, *ratio*, *uniform window*, *triangular window*, and *exponential window.* Each chunking mechanism is tested over a range of d values in order to make fair comparisons between chunking mechanisms, and to further the understanding of the behaviour of IDyOT across a range of parameters. Surface entropy, H_s, is the highest performing boundary strength measure from §10.6.3, and is retained for current experiment. The parametrisations for the experiment are summarised by Table 10-I.

Table 10-I: Parametrisations when testing chunking mechanisms in IDyOT.

Parameter	Set of values
Boundary strength measure	H_s
Chunking threshold	$0.0 \leq d \leq 8.0$
Chunking mechanism	$m \in \{absolute, delta, ratio, uniform\ window,$ $triangular\ window, exponential\ window\}$
Chunk alphabet size	100
Chunk equality viewpoint	`RootInt`⊗`ChordType`
Event bias	1
Chunk bias	1

10.6.4.1 Hypotheses

In general, the weighted window methods are expected to outperform the simpler *absolute*, *delta*, and *ratio* methods. In particular, the *triangular* and *exponentially* weighted windows are expected to be able to adapt naturally to the recent context of boundary strength measure values. The *absolute* chunking mechanism is expected to be the poorest performer, simply chunking whenever the boundary strength measure exceeds the threshold.

The previous comparison of boundary strength measures (§10.6.3, specifically Figure 10.6, bottom right) revealed an unexpected behaviour of IDyOT when employing surface entropy, H_s, as a boundary strength measure and *delta* as a chunking mechanism, whereby the optimal chunking in terms of overall mean information content occurs as d approaches 0. The current experiment enables a further exploration of this behaviour, in particular whether it holds over a variety of chunking mechanisms.

10.6.4.2 Results

Each chunking mechanism is tested across an initial pre-selected set of thresholds, d, before the results are observed and a second set of thresholds selected manually. The performance of each chunking mechanism is summarised in Figure 10.7, and given in more detail in Tables E-5, E-6, E-7, E-8, E-9, and E-10 (Appendix E). An initial cursory review of the results shows again that the results deviate substantially from the hypotheses above. Clearly, all of the weighted window methods fail to outperform both the *absolute* and *ratio* chunking mechanisms, both of which perform best as d approaches 0. In addition, the *absolute* chunking mechanism appears to outperform *delta*, and match *ratio*, although for reasons discussed below, this finding is somewhat trivial.

It is possible that the mean information content, $\bar{h}$, is primarily dependant on the mean chunk length, rather than more sophisticated higher order structure found in the upper layer (see Tables E-5, E-6, E-7, E-8, E-9, and E-10; Appendix E). In order to test this hypothesis, a multiple linear regression model tests the extent to which the performance, $\bar{h}$, can be accounted for by the independent variables of chunking mechanism, mean chunk length, and coverage, producing a model that accounts for 76.9% of the variance ($R^2 = 0.788$, $R^2_{adj} = 0.769$, $F(7, 77) = 40.88, p < 0.001$). The fit of the model is only marginally ($F(5) = 2.876$, $p = 0.020$) reduced if the chunking mechanism is removed as an independent variable ($R^2 = 0.743$, $R^2_{adj} = 0.742$, $F(2, 82) = 121.9, p < 0.001$). However, by further removing chunk coverage as an independent variable to leave only chunk length, the fit of the model is significantly ($F(1) = 138.85$, $p < 0.001$) reduced, accounting for only 31.4% of variance in $\bar{h}$ ($R^2 = 0.322$, $R^2_{adj} = 0.314$, $F(1, 83) = 39.46$, $p < 0.001$). Therefore, it appears that the mean chunk lengths and chunk coverage produced by IDyOT are more significant indicators of performance than the specific chunking mechanism employed.

The results indicate a substantial failing for this implementation of IDyOT: performance is optimal when the number of chunk boundaries is maximised. When $d = 0$ for the *absolute* and *ratio* mechanisms a chunk boundary is placed at every event. As `RootInt`$\otimes$`ChordType` is used for the chunk equality viewpoint, the first element of the current chunk, V, will always be undefined, and so $p({}_kE|{}_kV)$ removed from probability distribution combinations when predicting the surface event (see §10.3.4). The result is a pure surface system, identical to a first order IDyOM model taking two passes over the training set. Therefore, the finding that the *absolute* and *ratio* chunking mechanisms outperform *delta* and the weighted window methods is trivial, since for any method, d can be set to a (potentially negative) value that creates chunks at every event, and returns the lowest mean information content and best performance. To conclude, with the parametrisations tested so far, the impact of the upper layer has a detrimental impact on model performance.

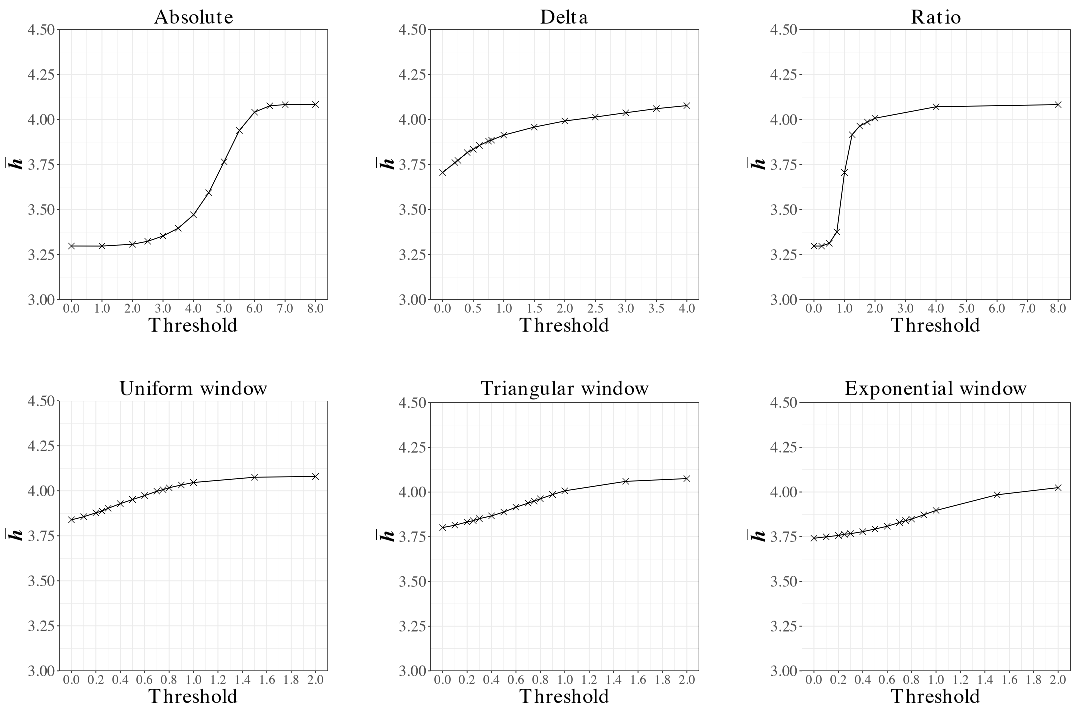


Figure 10.7: Comparative performance of IDyOT over six chunking mechanisms: *absolute*, *delta*, *ratio*, *uniform window*, *triangular window*, and *exponential window*.

10.6.5 Testing Chunk Equality Viewpoints

At this point the informal search procedure proposed in §10.6.1 has failed; the search finds a locally optimal parametrisation where the impact of the chunk layer has been nullified. However, other parametrisations lying outside the local parameter space searched may prove more fruitful, and so the search is continued with a marginal reset. A chunking mechanism of *delta* with a chunking threshold of $d = 1.0$ using surface entropy, H_s as the boundary strength measure produces musicologically meaningful segmentations, with a mean chunk length of 4.343 (Table E-6, Appendix E). These parameters are used for the following comparison of chunk equality viewpoints, instead of continuing with the optimal, albeit failed, parameters from §10.6.4.

The chunk equality viewpoint governs the labelling of chunks, such that two chunk sequences that have identical viewpoint sequences in the chunk equality viewpoint are given the same label (§10.3.4). Nine linked viewpoints are chosen for comparison, resulting from the cross product of three primitive viewpoints derived from `Root` (`Root`, `RootInt`, `ChromaDist`), and three from `ChordType` (`ChordType`, `FunctionType`, `MajType`). Each viewpoint provides a different level of abstraction, resulting from absolute against relative pitch representations and the categorisation of chord types. Preliminary runs show that the chunk equality viewpoint is highly dependent on the number of available chunk symbols in the chunk alphabet, $\mathcal{A}$. In order to minimise this effect the size of $\mathcal{A}$ is increased to 1,000 for the current experiment. The parametrisations of IDyOT used to compare chunk equality viewpoints are summarised in Table 10-J.

Table 10-J: Parametrisations when comparing chunk equality viewpoints in IDyOT.

Parameter	Set of values
Boundary strength measure	H_s
Chunking threshold	1.0
Chunking mechanism	*delta*
Chunk alphabet size	$1,000$
Chunk equality viewpoint	$\tau_{c_1} \otimes \tau_{c_2} \in \{$`Root, RootInt, ChromaDist`$\}\otimes$ $\{$`ChordType, FunctionType, MajType, Type`$\}$
Event bias	1
Chunk bias	1

10.6.5.1 Hypothesis

An optimal level of abstraction is expected that compromises between being too specific to generalise, and too general to make accurate predictions. As transpositionally equivalent sequences are musicologically considered almost identical, the optimal level of abstraction may be around that of `RootInt`. However, the findings of Chapter 8 concerning relative and absolute viewpoints may prove to apply to both surface and higher order structure, in which case a level of abstraction around `Root` would be expected.

10.6.5.2 Results

The results (tabulated in Table 10-K) conform to the hypothesis of a moderate optimal level of abstraction, with `RootInt`⊗`FunctionType` returning the lowest mean information content, $\bar{h}$, of 3.773 bits/symbol. This does not statistically significantly outperform the next best performing chunk equality viewpoint of `ChromaDist`⊗`FunctionType` (df = 347, t = 0.065, p = 0.474, Cohen's d = 0.000), but does significantly outperform the best chunk equality viewpoint using `Root`, which is `Root`⊗`FunctionType` (df = 347, t = 27.388, p < 0.001, Cohen's d = 0.240). Interestingly, both `Root` and `ChordType` components of the linked viewpoint show consistent signs of optimisation at moderate levels of abstraction, in other words, when `Root` is abstracted to `RootInt`, and `ChordType` to `FunctionType`.

Table 10-K: Comparative performance of chunk equality viewpoints in IDyOT.

Chunk equality viewpoints	$\bar{h}$
`Root`⊗`ChordType`	4.185
`Root`⊗`FunctionType`	4.007
`Root`⊗`MajType`	4.065
`RootInt`⊗`ChordType`	3.851
`RootInt`⊗`FunctionType`	3.773
`RootInt`⊗`MajType`	3.814
`ChromaDist`⊗`ChordType`	3.860
`ChromaDist`⊗`FunctionType`	3.778
`ChromaDist`⊗`MajType`	3.816

Note. Mean chunk length: 4.344, chunk coverage: 100%.

10.6.6 Testing Combination Biases

Two pairs of probability distributions must be combined when calculating the probability of the next surface event, $p({}_kE^i|{}_ke_1^{i-1}, u, {}_kV)$ (see §10.3.1.7). The current chunk is predicted by the surface context, $p({}_kV|{}_ke_1^{i-1})$, and the previous chunk, $p(V|u)$, to produce a single distribution $p({}_kV|{}_ke_1^{i-1}, u)$. The current surface event is similarly predicted by the surface context, $p(E^i|e_1^{i-1})$, and the current chunk, $p({}_kE^i|{}_kV)$ to produce

$p({}_kE^i|{}_ke_1^{i-1}, {}_kV)$. Probability distributions are combined with the geometrically weighted combination technique (Equation 3.7) as described in §3.4.4. Each combination is controlled by a chunk bias, b_c, for the current chunk combination, and an event bias, b_e, for the surface event combination. These determine how aggressively the combination is weighted towards the distribution with the lower relative entropy (Equation 3.10).

The optimal bias values cannot be determined prior to running, and so must be found empirically. In IDyOM, the LTM-STM and viewpoint biases can be found with an exhaustive search over all combinations of values (§5.7.3), although unfortunately such an approach is impractical due to the increased time complexity of IDyOT. Therefore, a greedy hill climbing search algorithm akin to the stepwise viewpoint selection algorithm (§3.5) is employed to find locally optimal values for the pair of biases. Bearing in mind that LTM-STM and viewpoint bias optimisation in previous research finds only a single minimum in the search space (see §5.7; Pearce, 2005; Whorley, 2013), the local minimum found by the search is likely to also be the global minimum. From initial starting values for b_c and b_e, at each iteration in the search, either bias may be increased or decreased by one, or remain the same. If each state at an iteration n is expressed as a tuple, $(b_c^n,\, b_e^n)$, then the set of possible states for the tuple (b_c^{n+1}, b_e^{n+1}) for the next iteration is:

$$\{(b_c^n + i, b_e^n + j) : -1 \leq i \leq 1, -1 \leq j \leq 1\}. \tag{10.24}$$

Alternatively, if possible combinations of b_c and b_e are arranged as a matrix with each cell representing a state, the next state may be any adjacent cell (including diagonals). At each step, the state that returns the lowest possible mean information content, $\bar{h}$, is selected for the next state, terminating when all possible next states return a higher $\bar{h}$. For the following search, initial values of $b_c = 1$ and $b_e = 1$ are used, with limits enforced that $0 \leq b_c \leq 8$ and $0 \leq b_e \leq 8$. The parameter subspace of IDyOT explored for the current experiment is given in Table 10-L, taking the best performing parameters from the previous experiments, although a non-optimal (see §10.6.4) chunking threshold of $d = 1.0$ is used to prevent the system chunking at every event. It is hoped that by finding a different set of chunk and event biases, the surface event predictions will benefit more from the upper layer, rather than hindering it as suggested by the fact the optimal chunking strategy places chunk boundaries at every event, effectively nullifying the predictions from the chunk layer (§10.6.5.2).

Table 10-L: Parametrisations when comparing chunk and event biases in IDyOT.

Parameter	Set of values
Boundary strength measure	H_s
Chunking threshold	1.0
Chunking mechanism	*delta*
Chunk alphabet size	100
Chunk equality viewpoint	`RootInt`$\otimes$`FunctionType`
Event bias	$0 \leq b_e \leq 8$
Chunk bias	$0 \leq b_c \leq 8$

10.6.6.1 Hypothesis

Concrete hypotheses are difficult to make for the current experiment: the reasons why one bias parameter outperforms another are not immediately apparent. In general, a high bias for a combination indicates that one of the probability distributions is both certain about the prediction (creating a low entropy distribution), and often successfully assigns a high probability to the predicted symbol. A low bias value may indicate a highly certain distribution that often incorrectly assigns a high probability to the wrong symbol. However, equally it may indicate that the distributions often have similar relative entropies, so there is no notable advantage in weighting towards the marginally more certain distribution. An informal observation of the relative entropies over all distributions during the runs of IDyOT so far reveals that for the prediction of the current chunk, the relative entropies of $p({}_kV|{}_ke_1^{i-1})$ and $p(V|u)$ are fairly even. However, when predicting the surface event, $p({}_kE^i|{}_kV_1^{i-1})$ is either much more or much less certain than, $p(E^i|e_1^{i-1})$; the two distributions are rarely equally certain. In this case, a high bias for b_e would indicate that when the predictions from the upper layer to the surface layer are certain they are usually correct, whilst a low bias would indicate that when the upper layer predictions for the surface event are certain they are often incorrect.

10.6.6.2 Results

The results of the search are given in Table 10-M, showing only the selected states of the greedy hill climbing algorithm.[17] The first point of note is that the final selected biases of $b_c = 0$ and $b_e = 8$ produce a mean information content of 3.487 bits/event, statistically significantly ($df = 347, t = 28.808, p < 0.001$, Cohen's $d = 0.514$) outperforming the initial biases of $b_c = 1$ and $b_e = 1$, which produce a mean information content of 3.816 bits/symbol. The search itself ran through seven iterations, first minimizing b_c, and then

[17] The full results for all states are given in Table E-11, Appendix E.

iteratively maximising b_e, producing a model where both parameters are at opposite extremes. Since the upper and lower bounds on the biases are arbitrary, a further analysis shows that $\bar{h}$ continues to decrease monotonically when $8 < b_e \leq 16$ (Table E-12, Appendix E). An additional run of the search algorithm with a starting state of $b_c = 5$, $b_e = 5$, further validates the methodology by converging to the same final state of $b_c = 0$, $b_e = 8$ (Table E-13, Appendix E).

Table 10-M: Selected chunk and event biases at each iteration of a greedy hill climbing algorithm.

Iteration	Chunk bias (b_c)	Event bias (b_e)	$\bar{h}$
start	1	1	3.816
1	0	2	3.699
2	0	3	3.628
3	0	4	3.581
4	0	5	3.546
5	0	6	3.521
6	0	7	3.502
7	0	8	3.487

The finding that the opposite extremes of both parameters are optimal is surprising, and unprecedented in multiple viewpoint research (see §5.7; Pearce, 2005; Whorley, 2013). The minimum chunk bias parameter suggests either that the distributions $p({}_kV|{}_ke_1^{i-1})$ and $p(V|u)$ have similar relative entropies, or that one makes certain, but incorrect predictions. Conversely, the maximised event bias parameter suggests when either $p({}_kE^i|{}_kV_1^{i-1})$ or $p(E^i|e_1^{i-1})$ are certain they make correct predictions. The surface event given the current chunk, $p({}_kE^i|{}_kV_1^{i-1})$, is simply predicted by converting the derived chunk viewpoint element at the current chunk index to the surface viewpoint elements. If this mapping is one-to-one the distribution would be nearly certain (bar smoothing), although for the current experiments, the mapping is one-to-many, as elements in `RootInt`$\otimes$`FunctionType` must be converted to `Root`$\otimes$`ChordType` (see §10.3.1.3). The fact that $p({}_kE^i|{}_kV_1^{i-1})$ often gives very specific predictions is an indication that at certain points in a sequence the system does benefit from the upper layer predictions. However, the performance of the best parametrisation of biases, 3.487 bits/symbol, is still unable to match the performance of IDyOT when chunks are created on every event (3.298 bits/symbol), nullifying the chunk layer (Figure 10.7). In conclusion, although the optimised chunk and event biases greatly improve predictive performance, they do not enable the prediction from the chunk layer to be integrated coherently into the surface layer.

10.6.7 Comparison with IDyOM

To summarise the previous experiments, the optimisation of parameters presented in §10.6.3, §10.6.4, §10.6.5, and §10.6.6 indicate strongly that the chunk layer in the current IDyOT implementation hinders rather than aids prediction. This is verified in a comparison of the best IDyOT parametrisation so far that still produces chunks (Table 10-N) against a surface-only model, essentially equivalent to an IDyOM model[18] that passes the training data twice. IDyOT returns a mean information content of 3.392 bits/symbol, which is statistically significantly ($df = 347, t = 19.798, p < 0.001$, Cohen's $d = 0.095$) outperformed by IDyOM's mean information content of 3.298 bits/symbol, although the effect size and absolute different in performance is small.

Table 10-N: Parametrisations and performance of IDyOT for comparison against IDyOM.

Parameter	Set of values
Boundary strength measure	H_s
Chunking threshold	1.0
Chunking mechanism	*delta*
Chunk alphabet size	$1,000$
Chunk equality viewpoint	`RootInt`$\otimes$`FunctionType`
Event bias	0
Chunk bias	16
Mean information content	3.392 bits/symbol
Chunk coverage	100%

A more detailed comparison of performance between the two models is made in Figure 10.8, plotting the information content, h, assigned to individual events by IDyOM and IDyOT, with the bottom-left to top-right diagonal signifying the point where one model outperforms the other. A high level of correlation is found between the models ($r = 0.986, p < 0.001$), although the slight asymmetry tending upward of the diagonal shows that IDyOM slightly outperforms IDyOT by a small amount over a large number of notes. An interesting cluster of events predicted with an h of between 2 and 4 by IDyOM, and around 1 by IDyOT goes against the trend of IDyOM outperforming IDyOT, and warrants further investigation.

A plausible explanation is found by observing that IDyOT outperforms IDyOM for events belonging to frequently occurring chunks (Figure 10.9). A distinct cluster of events where IDyOT outperforms IDyOM belong to chunks occurring just under 1,000 times within a cross-validation fold. Therefore, it appears the chunk layer of IDyOT is capable of improving predictions, but only in frequently occurring chunks.

[18] Using STMC1IU-LTM+C1I and bias weights of 2 and 1 for LTM-STM and viewpoint combination.

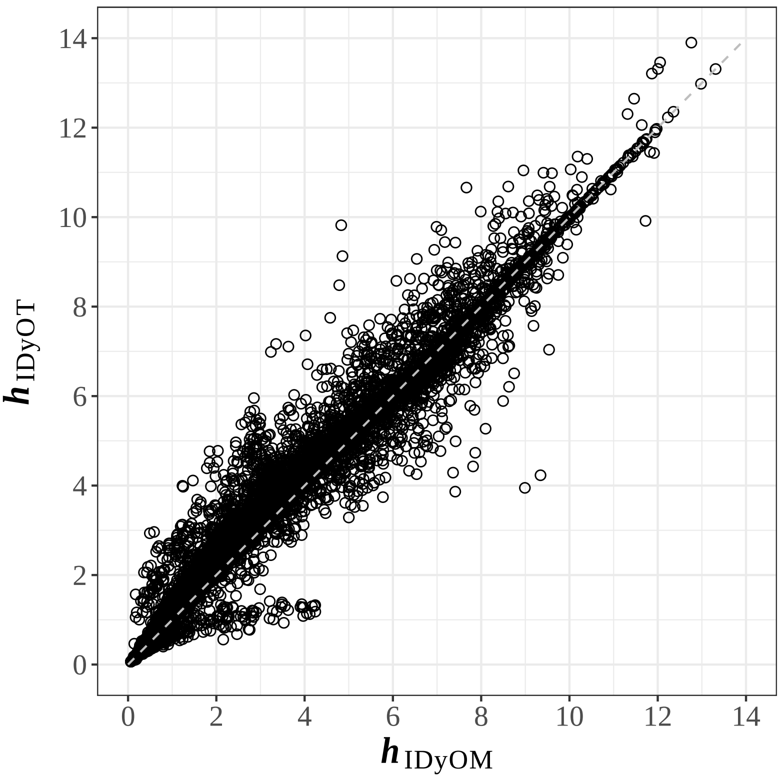


Figure 10.8: Information content, h, by event for IDyOT and IDyOM tested on a common corpus. Points above the dashed line $y = x$ indicate events predicted better by IDyOM, whilst points below are events better predicted by IDyOT.

Table 10-O gives further insight into these chunks, tabulating the 20 most frequent chunks stored in IDyOT's memory[19] after two epochs of the training data and one over the test data. The chunk sequence $[(\perp, m7), (5, 7)]$ corresponds with the cluster noted in Figure 10.9; a simple but vitally important chord progression in jazz more commonly known as $ii^7 - V^7$, or *two-five* (Levine, 1989, 1995). The top 20 ranked chunk sequences are musicologically meaningful jazz chord progressions; sequences include $[(\perp, m7), (5, 7), (5, M)]$, which is the full $ii^7 - V^7 - I$, $[(\perp, m7), (5, 7), (5, m7), (5, 7)]$, which is a section of a cycle of fifths, $[(\perp, M), (2, m7), (5, 7)]$, which is a common move away from the tonic, and $[(\perp, m7), (11, 7)]$, which is a tritone substitution.[20]

Reinforcing the findings in §10.6.2, the rank-frequency plot exhibits a strong ex-

[19] This is not the same as the most frequent chunks occurring in the corpus, since IDyOT forgets chunks that occur only once at the end of each epoch to create space in the chunk alphabet, $\mathcal{A}$ (see §10.6.2).

[20] $ii^7 - V^7 - I$ becomes $ii^7 - II\flat^7 - V$.

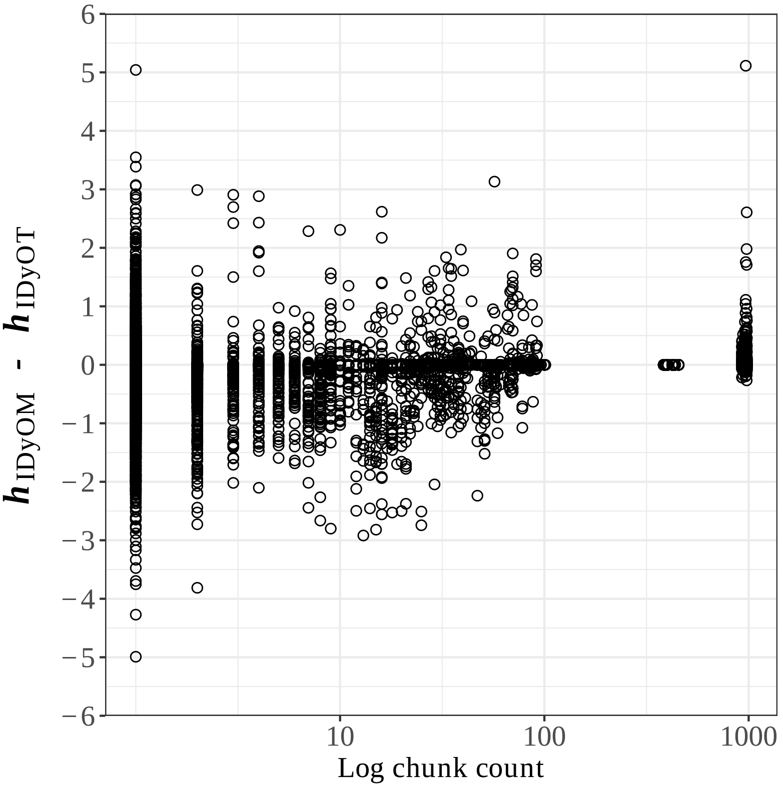


Figure 10.9: Scatter plot of performance difference ($h_{IDyOM} - h_{IDyOT}$) between IDyOT and IDyOM against chunk count (counted within each validation fold). Note that chunks that occur only once are not removed after the test phase, so appear in the plot.

ponential decay (Figure E-1, Appendix E), suggesting the highest ranked chunk types dominate the corpus and IDyOT's memory. Over a quarter (26.4%) of all chunks in memory have the type of the highest ranked chunk. The top 10 chunk types make up over half (52.5%) of all chunks in memory, and the top 100, 82.9%.

For events belonging to chunks that occur more than 900 times in memory, IDyOT significantly outperforms IDyOM by 0.047 bits/symbol ($df = 1157, t = 7.189, p < 0.001$, Cohen's $d = 0.021$). For chunks occurring more than 200 times, IDyOT outperforms IDyOM by 0.038 bits/symbol ($df = 1421, t = 7.160, p < 0.001$, Cohen's $d = 0.018$). For chunks occurring more than 20 times, the difference in performance is no longer statistically significant ($df = 2555, t = 1.257, p = 0.105$, Cohen's $d = 0.003$), but is still in favour of IDyOT. IDyOM only begins to outperform IDyOT when considering events from chunks occurring more than 16 times in memory, and for all subsequent

Table 10-O: Top 20 most frequent chunk sequences aggregated over cross-fold validation sets stored in IDyOT's memory.

Rank	Chunk sequence	Count	%
1	$[(\bot, m7), (5, 7)]$	9616	26.4%
2	$[(\bot, M)]$	4142	11.4%
3	$[(\bot, 7)]$	946	2.6%
4	$[(\bot, m7)]$	872	2.4%
5	$[(\bot, m7), (5, 7), (5, M)]$	860	2.4%
6	$[(\bot, M), (2, m7), (5, 7)]$	698	1.9%
7	$[(\bot, 7), (5, M)]$	587	1.6%
8	$[(\bot, m7), (11, 7)]$	516	1.4%
9	$[(\bot, m)]$	499	1.3%
10	$[(\bot, M), (9, m7), (5, m7), (5, 7)]$	409	1.1%
11	$[(\bot, M), (0, M)]$	404	1.1%
12	$[(\bot, 7), (0, m7), (5, 7)]$	342	0.9%
13	$[(\bot, M), (6, m7), (5, 7)]$	325	0.9%
14	$[(\bot, m7), (0, m7), (8, 7)]$	309	0.8%
15	$[(\bot, 7), (5, m7), (5, 7)]$	307	0.8%
16	$[(\bot, M), (1, m7), (5, 7)]$	287	0.8%
17	$[(\bot, 7), (11, 7)]$	286	0.8%
18	$[(\bot, m7), (5, 7), (5, m7), (5, 7)]$	279	0.8%
19	$[(\bot, 7), (0, 7)]$	277	0.8%
20	$[(\bot, M), (11, m7), (5, 7)]$	235	0.6%

Note. Chunks are viewpoint sequences in `RootInt`⊗`FunctionType`. For space, M represents the *tonic-major*; m the *tonic-minor*; 7 the *dominant*; and $m7$ the *pre-dominant.* % indicates the proportion of chunks in memory that have the same chunk sequence.

lower bounds from 16 and below. These findings reinforce empirically the earlier observation that IDyOT performs well for frequent chunks, but less so for infrequent chunks (Figure 10.9).

10.7 Conclusions and Discussion

This chapter has developed, presented, and tested a preliminary working implementation of the Information Dynamics of Thinking (IDyOT) cognitive architecture presented by Wiggins and Forth (2015) and Forth et al. (2016).[21] The implementation proposed by the current research is essentially a stratified DBN, with conditional probabilities taking on specific definitions due to relations between the so-called hidden states in the upper layer and the surface representation of the bottom layer. The implementation aims to

[21] Summarised in Chapter 9.

act as a minimally complete predictive model, containing all the components required to predict events on the surface layer by combining predictions of models from both surface and upper layers.

Using mean information content, $\bar{h}$, as a metric, empirical testing of the implementation aimed to find a locally optimal parametrisation, and to assess expected behaviours of the system. A comparison of boundary strength measures (§10.6.3) found that models that employed surface entropy, H, as a boundary strength measure performed better than those that employed information content, h, surface information content, h_s, or entropy, H. This contrasts to previous segmentation tasks (Griffiths et al., 2015; Pearce et al., 2010b; Wiggins, 2012a) where only h has been employed as a boundary strength measure. However, as the current study judged performance according to predictive power, rather than segmentation accuracy, H_s provides a noticeable advantage by being able to signify in advance whether the predicted event is the next event in the current chunk, or the first event of the next chunk. A multiple linear regression comparing chunk mechanisms (§10.6.4) showed a minimal effect of the six chunking mechanisms on performance, which instead was mainly accounted for by the mean chunk length, and the chunk coverage (the proportion of events in the testing sets contained within a chunk in memory). `RootInt`$\otimes$`FunctionType` proved the most effective chunk equality viewpoint (§10.6.5), an expected behaviour as it provides an appropriate level of abstraction for the chunk level. Surprisingly, an optimisation of the biases mediating the combinations of probability distributions predicting the current chunk, and the current surface event, place the event bias at the maximum possible value, and the chunk bias at the minimum. This suggests that when more certain predictions are made of the current surface event, E^i, by either the current chunk, V, or surface context, e_1^{i-1}, they turn out to be correct.

In all, the parameter search for IDyOT reveals substantial flaws in the choice of implementation. Of primary concern is the finding that the optimal chunk mechanism and threshold combination results in chunks being placed at every boundary, nullifying the effect of the upper layer on prediction. Indeed, a direct comparison (§10.6.7) between the best performing IDyOT parametrisation that still produces reasonable chunks, and an equivalent IDyOM model shows IDyOT is marginally, but statistically significantly, outperformed by 0.094 bits/event (3.392 bits/symbol to 3.298 bits/symbol). A detailed analysis of the performance of individual chunks showed that IDyOT outperformed IDyOM for the most common, musicologically meaningful chunks stored in memory, indicating that part of the system performs as expected. However, overall, it seems likely that the inability to generalise efficiently over the chunks learned causes IDyOT to make strong, but incorrect, predictions from the upper layer predicting the surface layer when the current chunk is moderately common or uncommon. As the results stand, there is no information theoretic justification for the addition of an upper chunk layer capturing higher order structure when tasked with predicting the surface layer. By extension, the current implementation does not support the core hypothesis of Wiggins and Forth (2015) that cognitive representations resulting from generators working at different temporal levels in IDyOT are formed primarily to make information theoretically efficient predictions.

There are a number of potential causes for the poor performance of this preliminary IDyOT implementation. Somewhat trivially, the informal search strategy used is only guaranteed to find a local minimum in the parametrisation space; there is potential for another parametrisation to produce a better mean information content that may outperform IDyOM. However, such an outcome seems relatively unlikely; largely representative proportions of the parameter space were explored and there appeared to be minimal cross-influence between subsets of parameters (for example, generally the chunk equality viewpoint is not influenced by the performance of the boundary strength measure). It is more likely that IDyOT's poor performance may be accounted for by the absence of some components of the cognitive architecture not present in the current implementation. In particular, the implementation of a geometric conceptual space (Gardenfors, 2000) representing chunks on the upper layers would greatly improve the architecture's ability to powerfully generalise over rare chunks encountered in the training data.

Chapter 11

Statistical Learning of Tonal Harmonic Structure

11.1 Overview

A modified implementation of IDyOT is developed and tested in this chapter. In an attempt to address some of the issues identified in Chapter 10, the implementation draws on tonal harmonic principles by labelling chunks with tonal centres rather than chunk symbols. Contrary to the predominantly bottom-up approach of the rest of the thesis, this approach uses some domain specific knowledge: specifically, in identifying a plausible tonal centre for the chord sequence of a given chunk. However, importantly, this domain specific knowledge may, in theory, be learned in a full IDyOT architecture with conceptual space representations (Gardenfors, 2000) for the upper layers. The anticipated advantages are twofold. Firstly, the finite chunk alphabet of arbitrary size can be reduced considerably, and in a meaningful way, to the set of 12 pitch classes. Secondly, the ability for the architecture to label chunks and their enclosed chords with tonal centres opens new avenues of evaluation relating to tonal harmonic analysis.

The chapter opens by introducing *relational viewpoints*, a new class of viewpoint necessary to relate the tonal centre to the surface predictions in the current implementation (§11.2). The modified implementation is described in §11.3, and its parameter space empirically tested in §11.4. §11.5 and §11.6 assess musicologically and empirically IDyOT's ability to describe tonal harmonic structure, before discussions and conclusions are drawn in §11.7.

11.2 Relational Viewpoints

A subtle extension to the multiple viewpoint framework as presented in Chapter 3 is required in order to develop a domain-specific implementation of IDyOT capable of capturing tonal harmonic structure. A defining feature of the established viewpoint classes (§3.3.2) is that only the partial function defining the viewpoint and the event sequence are required to create sequences of viewpoint elements, thus: $\Phi_\tau : \xi^* \rightharpoonup [\tau]^*$. The limitation of this compact description is that a viewpoint function may not use any information beyond the basic attributes of the surface event sequence. However, one of the key motivating factors driving the IDyOT cognitive architecture is that representations themselves may be (statistically) learned, defined, and deployed, in order to aid information theoretically efficient predictions of future events. As presented by Conklin and Witten (1995),[1] the multiple viewpoint framework is unable to capture this behaviour.

The proposed approach is to define a class of viewpoints that allow their viewpoint function to be defined not only from the preceding event sequence, $e_1^{i-1} \in \xi^*$, but also from an external (possibly statistical) model acting on the event sequence. Although the external model extracts information from the event sequence, the information extracted is not deterministic, potentially depending on random components or on an arbitrary training corpus. The proposed class of viewpoints are referred to as *relational viewpoints*, modelling a relation between the event sequence and an external component, and are signified by a type τ_l. A referent, $r \in \mathrm{P}$, is a symbol whose value is defined by an external model, drawn from a finite alphabet P. The referent may be used in the viewpoint function definition, Υ_{τ_l}, which takes as its argument a referent and the preceding event sequence to produce an element in the domain of the relational viewpoint, $[\tau_l]$. Table 11-A provides the type definitions for the functions required to map between the surface and viewpoint domains of relational viewpoints, providing the equivalent ordinary viewpoint functions for reference.

Table 11-A: Function type definitions of ordinary and relational viewpoints.

Description	Ordinary Viewpoint	Relational Viewpoint
Viewpoint function	$\Psi_\tau : \xi^* \rightharpoonup [\tau]$	$\Upsilon_{\tau_l} : \xi^* \times \mathrm{P} \rightharpoonup [\tau_l]$
Matching function	$\Phi_\tau : \xi^* \to [\tau]^*$	$\Omega_{\tau_l} : \xi^* \times \mathrm{P}^* \to [\tau_l]^*$
Inverse viewpoint function	$\Psi'_\tau : \xi^* \times [\tau] \to 2^{[\tau_b]}$	$\Upsilon'_{\tau_l} : \xi^* \times [\tau_l] \times \mathrm{P} \to 2^{[\tau_b]}$

`TonalInt`. A relational viewpoint is defined for the current task of capturing tonal harmonic structure. `TonalInt` functions similarly to `RootInt` or `RootIntFiP`, finding

[1] See also Pearce (2005), Whorley (2013), and the review provided in Chapter 3.

the chromatic interval class between the current root and a referent root, r (Equation 11.1). In this instance, the domain of r happens to be the domain of `Root`, so $\mathrm{P} = [\texttt{Root}]$. By way of example, for a sequence of roots, $\Phi_{\texttt{Root}}\left(e_1^5\right) = [0, 5, 1, 2, 7]$, and a referent of $r = 7$, $\Omega_{\texttt{TonalInt}}\left(e_1^5, 7\right) = [5, 10, 6, 7, 0]$. For the `TonalInt` viewpoint, r represents the tonal centre relevant to the sequence, inferred by an external statistical model (§11.3). This relational viewpoint differs from previous viewpoints used to capture structure in relation to a tonal centre (`cpintref` in Pearce, 2005, or `ScaleDegree` in Whorley, 2013), where the tonal centre (or referent pitch) is a basic attribute of the musical surface. The current approach with relational viewpoints offers two advantages in modelling cognitive processes. Firstly, the process of learning higher order representations, such as tonal harmony, can be modelled explicitly, rather than assuming they occupy a surface level of representation (see Cambouropoulos, 2010). Secondly, a relational viewpoint allows for cases where tonal centres change within a sequence, or are ambiguous at certain points in a sequence (see §9.5.2).

$$\Psi_{\texttt{TonalInt}}(e_1^i, r) = \begin{cases} \perp & \text{if } r = \perp \\ -1 & \text{else if } \Psi_{\texttt{Root}}(e_1^i) = -1 \\ -1 & \text{else if } r = -1 \\ \left(\Psi_{\texttt{Root}}(e_1^i) - r\right) \bmod 12 & \text{otherwise} \end{cases} \tag{11.1}$$

11.3 IDyOT as a Tonal Chunker

Motivated by the poor predictive performance of the IDyOT implementation presented in §10.3, the following section presents a modified version of IDyOT that aims to model tonal harmonic structure more explicitly. The implementation presented is referred to as a *Tonal Chunker*, as it seeks to find tonal harmonic structure through labelling chunks with a tonal centre, using the tonal information from the upper layer to inform surface predictions. The Tonal Chunker differs from the previous IDyOT implementation in a few key components. Firstly, chunks are not labelled according to a chunk equality viewpoint (as described in §10.3.4) but simply with a pitch class denoting a tonal centre. There is a subtle distinction to be made, which is that the chunk alphabet is no longer an arbitrary set of symbols mapping onto surface sequences, as is the case for $\mathcal{A}$ in §10.3.4, but is a viewpoint alphabet, namely [`Root`]. Therefore, the chunk alphabet of the IDyOT Tonal Chunker may have all of the mathematical and geometric properties of a pitch class representation, potentially enabling it to be modelled by `RootInt` on the upper layer. Sequences within chunks are not stored to memory as a sequence of elements in the viewpoint equality function, but as a sequence of elements in a relational viewpoint, namely `TonalInt`. Therefore, the probability calculations estimating the conditional probabilities of the current event given the current chunk, the current chunk given the surface context, and the current chunk given the previous chunk are redefined. The other components of the IDyOT implementation, specifically, the DBN architecture (§10.3.1), the chunk strength measure (§10.3.2), the chunking mechanism (§10.3.3), and the training procedure over epochs (§10.3.5), remain unchanged.

11.3.1 Labelling Chunks with Tonal Centres

The IDyOT Tonal Chunker implementation labels chunks with a tonal centre, denoted by a pitch class. The chunk alphabet is, therefore, equivalent to the domain of the `Root` viewpoint: $\mathcal{A} = [\texttt{Root}]$. Bearing in mind the hidden node ${}_kV$ holds the current chunk label (see Figure 10.2), the tonal centre can be described thus: ${}_kv \in [\texttt{Root}]$. Three *tonal centre viewpoints* potentially defining a tonal centre, ${}_kv$, from a sequence of chunk events, e_s^i, are proposed and later tested. $\Psi_{\texttt{FirstRoot}}(e_1^i) = \Psi_{\texttt{Root}}(e^1)$ selects the first root of the sequence, while $\Psi_{\texttt{LastRoot}}(e_1^i) = \Psi_{\texttt{Root}}(e^i)$ selects the last. $\Psi_{\texttt{LastRoot}}$ is modified, noting that if the final chord fulfils a dominant function (contains a minor 7th), its root is more likely to be on the fifth scale degree (V), therefore placing the tonal centre a perfect 5th (7 semitones) below. The modified viewpoint is named `LastRootTonal`, defined as:

$$\Psi_{\texttt{LastRootTonal}}(e_1^i) = \begin{cases} \big(\Psi_{\texttt{Root}}(e^i) - 7\big) \bmod 12 & \text{if } \Psi_{\texttt{FunctionType}(e_1^i)} = \textit{dominant} \\ \Psi_{\texttt{Root}}(e^i) & \text{otherwise} \end{cases} \tag{11.2}$$

The three viewpoints have varying degrees of musicological validity, with both `LastRoot` and `LastRootTonal` taking advantage of the tendency for harmonic chunks to end in a perfect cadence so the final root is often the tonic. However, `FirstRoot` has the computational advantage of being defined from the start of a chunk, allowing the hidden node ${}_kV$ to always be defined, and never need be summed out as a hidden variable (§10.3.1.7). Like many of the viewpoints presented in this research, this rigid rule-based approach is not expected to give an entirely accurate account of tonal harmonic structure, but instead attempts to find statistical structure by noting a general tendency. An empirical comparison of these viewpoint is conducted in §11.4.

Where sequences of events inside a chunk in the previous implementation (see §10.3.4) were stored in the statistical model of the upper layer as elements in the chunk equality viewpoint, $\tau_{c_1} \otimes ... \otimes \tau_{c_n}$, a *tonal chunk viewpoint* is used to store such sequences in the current Tonal Chunker implementation. The tonal chunk viewpoint is any number of linked or merged viewpoints, including the `TonalInt` viewpoint: $\tau_{c_1} \otimes ... \otimes \tau_{c_n} \otimes \texttt{TonalInt}$. Overall, the tonal chunk viewpoint must fully predict the target surface attributes. In a more general implementation the viewpoint would comprise a number of linked or merged viewpoints, and any relational viewpoint where the referent, r, labels chunks and the viewpoint itself predicts one of the surface basic attributes.

11.3.2 Probability of Current Chunk given Previous Chunk

As chunk labels now refer to specific `Root` elements, rather than arbitrary sequences (see §10.3.4), a viewpoint approach to predictions may be applied to the upper layer when estimating $p(V|u)$. The current chunk on the upper layer is predicted with a `Root` viewpoint using a first-order Markov model employing interpolated smoothing (Equation 5.2) and escape method C (Table 5-A). The `Root` viewpoint on the upper layer is separate from any surface `Root` viewpoints in terms of the associated statistical models and seen alphabets. Alternatively, any other viewpoint derived from `Root` could model the chunk layer. However, since `Root` is found to consistently outperform its derived viewpoints (see §5.7.3, §6.4, §7.5, and §8.4.1) it is chosen in preference to, for example, `RootInt`.

11.3.3 Probability of Current Chunk Centre given Surface Context

The probability of the tonal centre of the current chunk given the surface context is denoted by $p({}_kV|e_1^{i-1})$. Intuitively, as the surface events since the last chunk boundary are processed, a probability estimate of the most likely tonal centre can be estimated by converting the surface roots to scale degrees with `TonalInt`, and comparing the resulting sequence to the sequences of `TonalInt` stored in the statistical models of the upper layer. More formally, the prediction is a maximum likelihood estimate, using interpolated smoothing (Equation 5.2) and escape method C (Table 5-A). Let e_{i-k+1}^{i-1} be the surface events of an ongoing chunk, indexed by the event index i, and the chunk index k. If V is the current chunk symbol (also denoting the tonal centre), $c_c(\Omega_{\texttt{TonalInt}}(e_{i-k+1}^{i-1}, {}_kV))$ is the count of chunk sequences stored in the upper layer.[2] The type count $t_c(e_{i-k+1}^{i-1})$ is given in Equation 11.3 for clarity, followed by $p({}_kV|e_1^{i-1})$ in Equation 11.4.

[2] Recall that the specialised counting function $c_c(e_x^y)$ matches chunk boundaries as well as viewpoint elements, therefore sequences of events which cross chunk boundaries will not be counted. Only sequences starting on the first event of a chunk are included in the counts.

$$t_c\left(e_{i-k+1}^{i-1}\right) = \left|\left\{v \in \mathcal{A} : c_c\left(\Omega_{\texttt{TonalInt}}\left(e_{i-k+1}^{i-1}, v\right)\right) > 0\right\}\right| \tag{11.3}$$

$$\begin{aligned} p\left({}_kV|e_1^{i-1}\right) = & \frac{c_c\left(\Omega_{\texttt{TonalInt}}\left(e_{i-k+1}^{i-1}, {}_kV\right)\right)}{\sum_{{}_kv\in\mathcal{A}} c_c\left(\Omega_{\texttt{TonalInt}}\left(e_{i-k+1}^{i-1}, {}_kv\right)\right) + t_c\left(e_{i-k+1}^{i-1}\right)} + \\ & \frac{t_c\left(e_{i-k+1}^{i-1}\right)}{\sum_{{}_kv\in\mathcal{A}} c_c\left(\Omega_{\texttt{TonalInt}}\left({}_ke_{i-k+1}^{i-1}, {}_kv\right)\right) + t_c\left(e_{i-k+1}^{i-1}\right)} \times \\ & \frac{1}{|\mathcal{A}| - |\mathcal{A}'| + 1} \end{aligned} \tag{11.4}$$

11.3.4 Probability of Current Surface Event Given the Current Chunk

Similarly, $p({}_kE|{}_kV)$, the prediction of the current event given the tonal centre of the current chunk, must also be redefined. The tonal centre of the current chunk acts as a relative anchor in predicting the current surface event, allowing predictions to be made in the form of tonal chord functions, such as tonic, dominant, and subdominant.[3] Let τ_c represent the tonal chunk viewpoint, $\tau_{c_1}\otimes ... \otimes\tau_{c_n}\otimes\tau_{\texttt{TonalInt}}$. Like the previous IDyOT implementation, the prediction is made over the tonal chunk viewpoint, predicting $p({}_kT|{}_k^jV)$ where ${}_kT \in [\tau_c]$, before using Υ'_{τ_c} to convert back to the basic attributes of the surface viewpoint. Unlike the previous IDyOT implementation, ${}_kV$ only contains information on the tonal centre and the chunk index; there is no mapping onto a chunk sequence. Therefore, as there is no information available on the elements of the current chunk, counts are made of all possible sequences of chunk elements seen so far that match the current chunk element. To clarify, if $x_1^k \in [\tau_c]^*$ is a sequence of chunk elements of length k, $c_c(\Upsilon_{\tau_c}(x_1^k, {}_kV) = {}_kT)$ is the number of times the chunk element T has been seen, given an arbitrary sequence, x_1^k, and a chunk symbol ${}_kV$ (containing the tonal centre and chunk index k). Equation 11.5 gives the type count, $t_c({}_kV)$ of a tonal centre at a chunk index k, and Equation 11.6 the probability of a chunk element ${}_kT$ at a chunk index k, given the chunk symbol ${}_kV$. It is worth noting that in practice x_1^k need not be exhaustively expanded to include all sequences in $[\tau_c]^*$, but may simply be all sequence of length k seen by the model, easily found by linking sibling nodes in the prefix tree (see Figure 11.1).

[3] For example, a dominant chord would be a `TonalInt` element of 7, potentially produced by $\Upsilon_{\texttt{TonaInt}}(8, 1) = 7$.

$$t_c\left({}_kV\right) = \left|\left\{x_1^k \in [\tau_c]^* : c_c\left(\Omega_{\tau_c}\left(x_1^k, {}_kV\right)\right) > 0\right\}\right| \tag{11.5}$$

$$\begin{aligned} p\left({}_kT|{}_kV\right) = & \frac{\sum_{x_1^k \in \xi^*} c_c\left(\Upsilon_{\tau_c}\left(x_1^k, {}_kV\right) = {}_kT\right)}{\sum_{{}_kt \in [\tau_c]} \sum_{x_1^k \in [\tau_c]^*} c_c\left(\Upsilon_{\tau_c}\left(x_1^k, {}_kV\right) = {}_kt\right) + t_c\left({}_kV\right)} + \\ & \frac{t_c\left({}_kV\right)}{\sum_{{}_kt \in [\tau_c]} \sum_{x_1^k \in \xi^*} c_c\left(\Upsilon_{\tau_c}\left(x_1^k, {}_kV\right) = {}_kt\right) + t_c\left({}_kV\right)} \times \\ & \frac{1}{|\mathcal{A}| - |\mathcal{A}'| + 1} \end{aligned} \tag{11.6}$$

11.3.5 Tonal Chunker Prediction Illustrations

A few illustrative predictions are made from a ‘pen-and-paper’ example in a similar manner to §10.4. Again, maximum likelihood estimations are simplified considerably by removing the smoothing elements:

$$p({}_kV|e_1^{i-1}) = \frac{c_c(\Omega_{\texttt{TonalInt}}(e_{i-k+1}^{i-1}, {}_kV))}{\sum_{{}_kv \in \mathcal{A}} c_c(\Omega_{\texttt{TonalInt}}(e_{i-k+1}^{i-1}, {}_kv))} \tag{11.7}$$

$$p({}_kT|{}_kV) = \frac{\sum_{x_1^k \in \xi^*} c_c(\Upsilon_{\tau_c}(x_1^k, {}_kV) = {}_kT)}{\sum_{{}_kt \in [\tau_c]} \sum_{x_1^k \in [\tau_c]^*} c_c(\Upsilon_{\tau_c}(x_1^k, {}_kV) = {}_kt)}. \tag{11.8}$$

Here, a tonal chunk viewpoint, τ_c, of `TonalInt` is used to simply model the root progressions of an imaginary sequence. The predictions are made on a partly trained model, which has so far learned the chunks shown in Table 11-B. The chunks are added to a prefix tree (Figure 11.1) as they are processed in an online manner. Each node is labelled with a node count, the number of times a chunk containing the chunk element passes that node. Any path from the root ($) to a node where the node count is less than the sum of the node counts of its parent nodes is a complete chunk sequence. If the node count is equal to the sum of the node counts of the parent nodes, the sequence is an incomplete chunk (a prefix).

Table 11-B: Frequency counts of chunks used in a 'pen and paper' example.

Chord function elements	Chunk elements (`TonalInt`)	Count
$[ii, V]$	$[2, 7]$	5
$[ii, V, I]$	$[2, 7, 0]$	4
$[ii, ii\flat, I]$	$[2, 1, 0]$	1
$[V, I]$	$[7, 0]$	3
$[V]$	$[7]$	4
$[V, V]$	$[7, 7]$	2
$[V, V, V]$	$[7, 7, 7]$	1

For the first illustrative example, given the sequence of `Root` elements $e_{k-i+1}^{i-1} = [4, 9]$ and the chunk index $k = 2$, the goal is to estimate the probability, $p\left({}_2v = 2|e_1^{i-1}\right)$, that the current chunk symbol (the tonal centre), ${}_2v$ is 2. The chunk element sequence $\Omega_{\texttt{TonalInt}}(e_1^2 = [4, 9], {}_2v = 2)$ is $[2, 7]$, matched 9 times in the prefix tree: $c_c([2, 7]) = 9$. Of all other ${}_2v \in \mathcal{A}$, only ${}_2v = 9$ can be matched from the surface context, producing a chunk element sequence of $\Omega_{\texttt{TonalInt}}(e_1^2 = [4, 9], {}_2v = 9) = [7, 0]$, occurring 3 times. Therefore, $p\left({}_kv = 2|e_1^{i-1}\right) = \frac{9}{9+3} = \frac{3}{4}$.

The second example estimates the probability of surface symbol ${}_2E = 8$ at a chunk index of $k = 2$, given a chunk symbol ${}_kV = 1$, alternatively expressed as $p({}_kE = 8|{}_2V = 1)$. First, $p({}_kT = 7|{}_2V = 1)$ is calculated, following the sibling nodes at a depth of $k = 2$ in the prefix tree this gives $\frac{9+3}{9+3+3+1} = \frac{3}{4}$. Applying $\Upsilon'_{\texttt{TonalInt}}(7, 1)$ gives a surface element of 8, as this is a one-to-one mapping then $p({}_kE = 8|{}_2V = 1) = \frac{3}{4}$.

11.4 Testing Tonal Chunker Parametrisations

The parameter space of the IDyOT Tonal Chunker is explored in a similar manner to the initial implementation in Chapter 10. As before, the search gives an empirical overview of the predictive performance of the implementation, allows a potential optimal parametrisation to be established, and gives an insight into the behaviour of some of the components of the cognitive architecture. The informal search strategy as described in §10.6.1 is deployed, searching subspaces of the parameters, carrying over optimal sub-parametrisations to the next subspace search, using mean information content, $\bar{h}$, as a heuristic. A STMC1IUM-LTM+C1IM model[4] using bias weights of $b = 2$ and $b = 1$

[4] See §5.2.1.4 for details on the shorthand model notation.

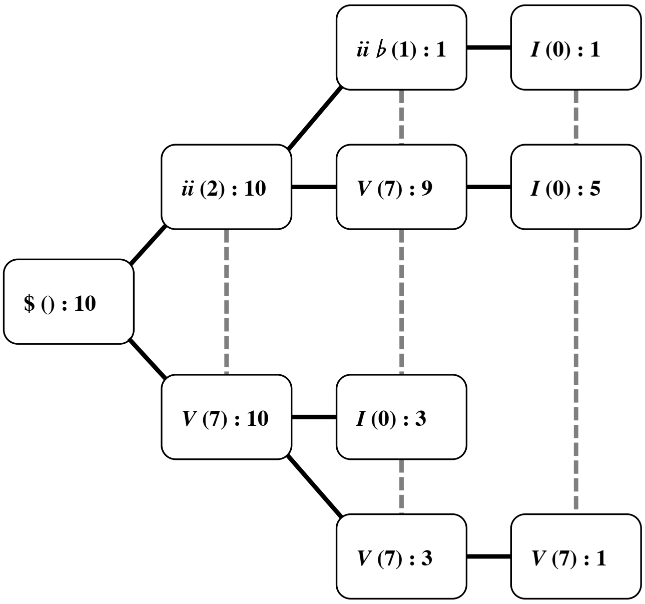


Figure 11.1: Prefix tree of tonal chunk sequences in a 'pen-and-paper' example for the IDyOT Tonal Chunker. Each node is given a label in the form: <*chord function*> (<*chunk element*>) : <*count*>. Bold lines indicate parent-child edges, and dashed lines indicate false edges between siblings.

for LTM-STM and viewpoint combination predicts the surface layer, with the *Real Book Vol. 1* (Table 4-E, dataset 1) used as the training and testing corpus.

The parameter space explored is shown in Table 11-C. In order, the tonal centre viewpoint, tonal chunk viewpoint, combination biases, and chunking thresholds are optimised. The chunk strength measure and chunking mechanism are not altered for the current implementation, and so the best performing parameters from §10.6.3 and §10.6.4 are retained.

11.4.1 Testing Tonal Chunk and Tonal Centre Viewpoints

Together, the tonal chunk and tonal centre viewpoints define how chunk labels and chunk sequences are stored in memory (§11.3.1). Chunk sequences consist of sequences of elements in the tonal chunk viewpoint; $\Psi_{\tau_{c_1} \otimes \tau_{c_2}}(e^i_s)$. The tonal chunk viewpoints tested are `TonalInt`⊗`ChordType`, `TonalInt`⊗`FunctionType`, and `TonalInt`⊗`MajType`, offering different levels of abstraction for chunk sequences to be stored. Given the high performance of `FunctionType` in the previous IDyOT implementation (§10.6.5), `TonalInt`⊗`FunctionType` is expected to perform the best of the three viewpoints. The

Table 11-C: Parameter space explored for the IDyOT Tonal Chunker.

Parameter	Set of values
Chunk strength measure	H_s
Chunking threshold	$0.0 \leq d \leq 4.0$
Chunking mechanism	*ratio*
Chunk alphabet size	$\lvert[\texttt{Root}]\rvert$
Tonal centre viewpoint	$\tau_e \in$ $\{\texttt{FirstRoot}, \texttt{LastRoot}, \texttt{LastRootTonal}\}$
Tonal chunk viewpoint	$\tau_{c_1} \otimes \tau_{c_2} \in$ $\texttt{TonalInt} \otimes \{\texttt{ChordType}, \texttt{FunctionType}, \texttt{MajType}\}$
Event bias	$0 \leq b_e \leq 8$
Chunk bias	$0 \leq b_c \leq 8$

tonal centre viewpoint labels a chunk with a tonal centre by considering either the first root of the chunk (`FirstRoot`), the last root of the chunk (`LastRoot`), or the last chord type and root (`LastRootTonal`). Of the three viewpoints, `LastRootTonal` is expected to perform best as it differentiates between chord types typically associated with dominant functions and ones associated with tonic functions, using this knowledge to place the tonal centre on the final root, or a perfect 5th below.

Tonal chunk viewpoints are tested first whilst keeping the tonal centre viewpoint fixed to `LastRootTonal`, before retaining the best tonal chunk viewpoint and comparing tonal centre viewpoints (Table 11-D). The chunking threshold is fixed at 1.0, and the chunk and event biases at 0 and 8 respectively. Surprisingly, the tonal chunk viewpoint with the highest level of abstraction, $\texttt{TonalInt}\otimes\texttt{MajType}$, statistically significantly ($df = 347$, $t = 29.629$, $p < 0.001$, Cohen's $d = 0.110$) outperforms the more moderate level of abstraction $\texttt{TonalInt}\otimes\texttt{FunctionType}$, although the absolute difference and effect size are small. However, as predicted, `LastRootTonal` is the best performing tonal centre viewpoint, statistically significantly ($df = 347$, $t = 34.282$, $p < 0.001$, Cohen's $d = 0.066$) outperforming `LastRoot`, although again with a small absolute difference and effect size. Noting that the final root of a chunk is a future event whilst a chunk is ongoing, this result is meaningful in the context of modelling higher order structure as it suggests that at least some information from potential events in the future influence the current event. By contrast, even though `FirstRoot` is defined at all points except the first event of a chunk, it does not improve the predictive performance of the model. These findings are very much in line with Western tonal harmonic theory, where tonal centres are usually defined by cadential figures at the ends of phrases.

Table 11-D: Comparing the performance of tonal chunk and tonal centre viewpoints in the IDyOT Tonal Chunker.

Tonal chunk viewpoint	Tonal centre viewpoint	$\bar{h}$
TonalInt⊗ChordType	LastRootTonal	3.869
TonalInt⊗FunctionType	LastRootTonal	3.872
TonalInt⊗MajType	LastRootTonal	3.754
TonalInt⊗MajType	FirstRoot	4.371
TonalInt⊗MajType	LastRoot	3.823
TonalInt⊗MajType	LastRootTonal	3.754

Note. The upper half of the table first compares tonal chunk viewpoints, before the bottom half compares tonal centre viewpoints.

11.4.2 Testing Chunk and Event Combination Biases

The chunk bias, b_c, and event bias, b_e, control how aggressively the distribution combinations of $\varsigma(p(E^i|e_1^{i-1}), p({}_kE^i|{}_kv))$ and $\varsigma(p({}_kv|{}_ke_1^{i-1}), p({}_kv|u))$ are weighted towards the distribution with the lowest relative entropy (see §3.4.4 and §10.3.1.7).[5] The greedy hill climbing search previous employed in §10.6.6 searches the space of the Cartesian product of b_c and b_e. The IDyOT implementation of §10.3 maximised the event bias, whilst minimising the chunk bias. However, as the Tonal Chunker implementation of IDyOT handles chunk memory and predictions from the upper layer in a different manner it is possible that different biases will be found.

Table 11-E shows the selected states of the search, initialised at $b_c = 1$, $b_e = 1$, and terminating at $b_c = 8$, $b_e = 8$.[6] The final state of the search is found to be a statistically significant ($df = 347$, $t = 21.564$, $p < 0.001$, Cohen's $d = 0.254$) improvement over the initial state, with a reasonable absolute difference in $\bar{h}$ of 0.277 bits/symbol. The maximised biases for both chunk and event combination suggests that, for both components of the system, the least uncertain distribution usually makes the most accurate predictions.

11.4.3 Testing Chunking Thresholds

The previous two experiments fix the chunking threshold, d, at 1.0. As the chunk strength measure (§10.3.2) is the surface entropy, H_m, and the chunking mechanism (§10.3.3) is the *ratio* method, the chunking strategy simply places a chunk boundary at every event where the surface entropy of the current event's context is larger than that

[5] Recall that ς is a function that combines probability distribution over the same alphabet, as described in §3.4.4.

[6] The chunking threshold is fixed at 1.0.

Table 11-E: Selected chunk and event biases at each iteration of a greedy hill climbing algorithm in the IDyOT Tonal Chunker.

Iteration	Chunk bias (b_c)	Event bias (b_e)	$\bar{h}$
start	1	1	4.014
1	2	2	3.912
2	3	3	3.851
3	4	4	3.811
4	5	5	3.783
5	6	6	3.763
6	7	7	3.748
7	8	8	3.737

of the previous event. The final parametrisation test explores the impact of varying the chunking threshold, controlling how often chunk boundaries are placed by IDyOT. An expected behaviour would be for the cognitive architecture to find a balance between chunking too often (creating small chunks that do not generalise the surface data) and chunking too infrequently (creating long chunks resulting in sparse statistical models).

The results of varying d over the set $\{0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 3.0, 4.0\}$ are shown in Figure 11.2. It is immediately apparent that no compromise between the two chunking extremes is found, with predictive performance optimized when the d is maximised. Only 462 chunk boundaries are found in the test sets when $d = 4.0$, and therefore, bearing in mind that there are only 348 pieces in the corpus, it can be safely assumed that $\bar{h}$ will remain at around 3.474 bits/symbol if d is further increased. Although the finding that a minimal segmentation optimises $\bar{h}$ is the opposite to the first IDyOT implementation, where the optimal model had a maximal segmentation (§10.6.4), the overall effect is the same. Both extremes nullify the impact of the upper layer, in the maximal segmentation case in the preliminary IDyOT implementation `RootInt` is undefined for the first event of each segment, and in the minimal segmentation case in the current implementation the chunks are too long to be matched against in a very sparse statistical model.

The conclusions of the parametrisation of the IDyOT Tonal Chunker are, therefore, similar to the first IDyOT implementation of Chapter 10. The cognitive architecture does not exhibit expected information theoretic behaviours, finding parametrisations that nullify the impact of the upper layer. With Table 11-F as a guide, a musically meaningful mean chunk length of between two and five events (c.f. Lerdahl & Jackendoff, 1983; Marsden, 2010; Pachet, 2000; Rohrmeier, 2011; Steedman, 1984; Ulrich, 1977) would require an optimum in $\bar{h}$ to be around $1.0 \leq d \leq 1.3$.

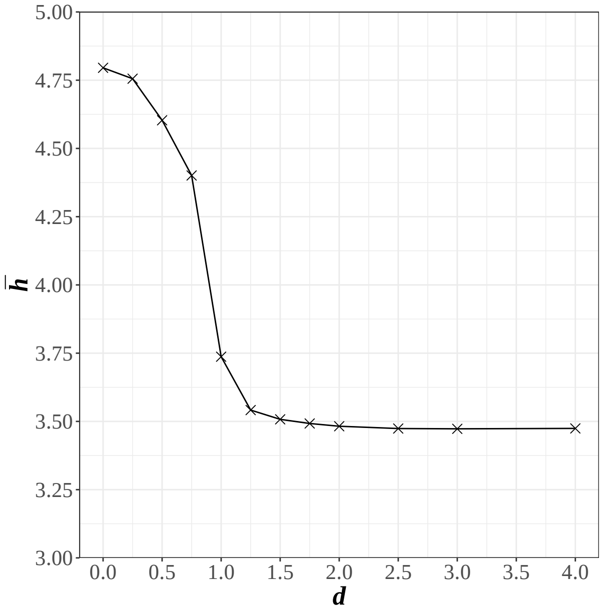


Figure 11.2: The mean information content, $\bar{h}$, of the IDyOT Tonal Chunker across a range of chunking thresholds, d.

Table 11-F: Predictive performance, $\bar{h}$, and mean chunk lengths resulting from different chunking threshold in the IDyOT Tonal Chunker.

d	Mean chunk length	$\bar{h}$
0.00	1.000	4.795
0.25	1.010	4.755
0.50	1.090	4.603
0.75	1.224	4.400
1.00	2.139	3.737
1.25	4.311	3.542
1.50	6.311	3.508
1.75	8.032	3.492
2.00	10.098	3.482
2.50	14.972	3.474
3.00	20.818	3.473
4.00	32.752	3.474

11.5 Musicological Analyses

So far, IDyOT has primarily been evaluated in its ability to compress information as measured by mean information content (§10.6 and §11.4.2), alongside some observations on its information theoretic behaviour. However, this approach neglects to evaluate the cognitive architecture in its ability to predict human behaviour in cognitive and perceptual tasks such as musical expectation, memory, and perception. At this early stage in the development of IDyOT, a full battery of human behavioural studies are deferred until the clear flaws in the current implementations have been addressed. However, as an explanatory model of tonal harmonic cognition, the IDyOT Tonal Chunker can be evaluated against descriptive models of tonal harmony. Following Wiggins et al. (2010) and Wiggins (2012b), music theory and musicology can be viewed as sophisticated descriptive models of music cognition, where structural analysis, tonal harmonic theory, and melodic motivic analysis can be viewed as proxies for internal, often difficult to observe, cognitive processes.

The following sections evaluate the IDyOT Tonal Chunker in its ability to produce musically meaningful tonal harmonic analyses of two jazz standards. The analysis is roughly equivalent to Riemannian analysis (Riemann, 1895) in identifying scale degrees of chords according to local tonal centres, and in general does not extend to the analysis of prolongation or dependencies (Lerdahl & Jackendoff, 1983; Schenker, 1979), although some indication is given by the probabilistic influences between consecutive symbols on each level. In contrast to the previous evaluations, the following musicological analyses offer a detailed, small-scale assessment of the behaviour of the cognitive architecture. The two selected jazz standards are handled reasonably by IDyOT, and are chosen for their potential to produce harmonically interesting analyses, rather than being a strict representation of IDyOT's performance.[7]

11.5.1 *"Solar"* by Miles Davis

"Solar" by Miles Davis takes a modified 12-bar blues form in C minor. The essence of the blues structure is captured by IDyOT (Figure 11.3) by modulating to the subdominant (F major) in bar 5, with a return to the tonic minor in the closing turnaround. Harmonically, the most interesting passage of *"Solar"* is the journey from subdominant (F) to tonic (C) in bars 5-12. A harmonic analysis according to standard jazz conventions (e.g., Levine, 1989, 1995) would interpret this as three $ii - V - I$ shapes cadencing in $B\flat$, $D\flat$, and C, using parallel major-to-minor movement to transition between the first two ($FM - Fm^7$, $E\flat M - E\flat m^7$), and chromatic voice leading to move to the D half-diminished chord for the third ($D\flat M - D\varnothing$). In labelling chords and tonal centres, IDyOT's interpretation is almost perfect. One exception occurs when interpreting the FM in bar 6 as II of $E\flat$ rather than I of F, arising from grouping the FM with the following chunk, rather than

[7] §11.6 addresses this issue with a broader evaluation of IDyOT's performance as a tonal harmonic segmenting and labelling tool.

the preceding. One might argue that the analysis is over segmented, dividing $ii - V - I$ chunks into two separate units. However, bearing in mind that the internal transitional probabilities on both levels within these chunks are high (signified by bold arrows), if the hierarchical process of chunking and labelling were to be applied recursively upwards forming higher layers, the two separate chunks would immediately be joined. The issue, therefore, is at the level the segmentation is carried out, rather than the segmentation *per se*. Finally, the opening chunk is mislabelled as G rather than C, as a result of using the final chord to identify the tonal centre.[8] An approach that blended both `FirstRoot` and `LastRootTonal` tonal centre viewpoints would be necessary to correctly label every chunk in this piece.

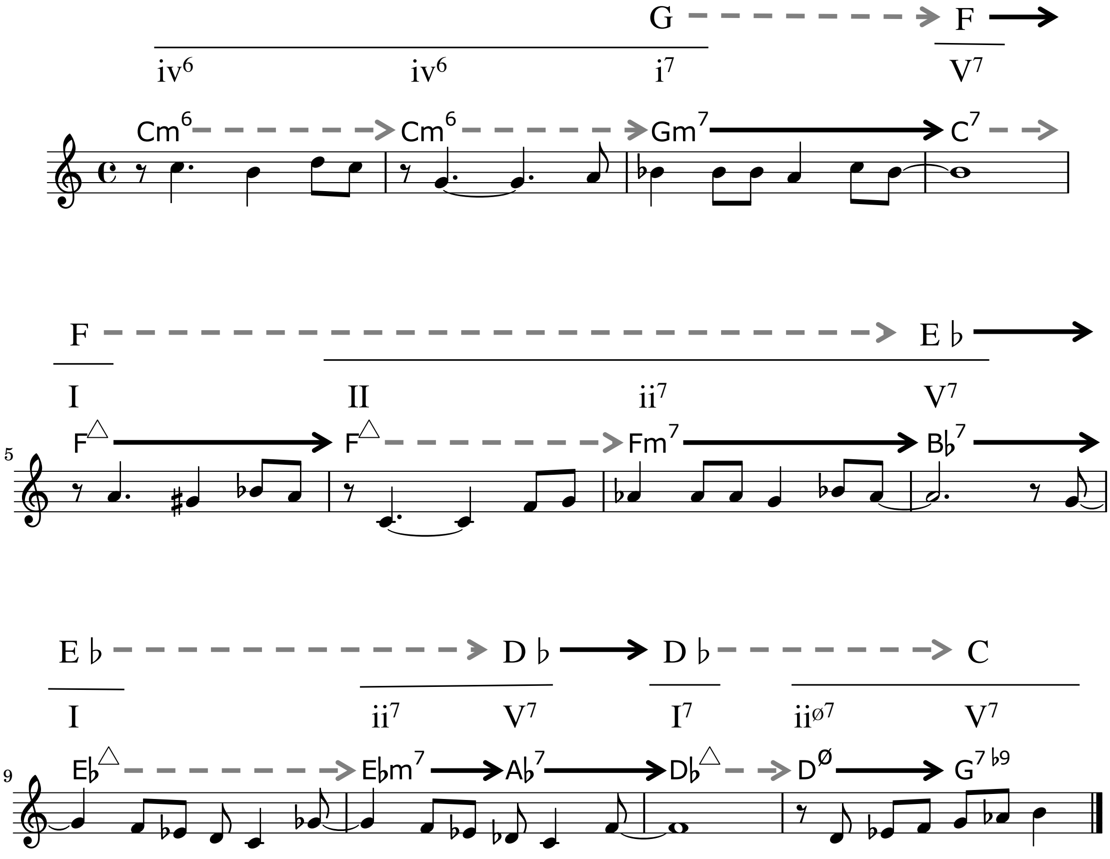


Figure 11.3: IDyOT parsing of *"Solar"* by Miles Davis. Local tonal centres spanning chunks (plain horizontal lines) are shown on the top line, which are used to inform scale degree on the second line. On both levels, high probability transitions are indicated with bold arrows, and low probability with grey.

[8] Interestingly, the opening is also incorrectly identified by the system of Pachet (2000) as in $B\flat$ major, suggesting it is a challenge to interpret for computational models in general.

11.5.2 *"Giant Steps"* by John Coltrane

"Giant Steps" by John Coltrane is the epitome of the 'Coltrane Changes' approach to jazz harmony (H. Martin, 2012; Waters, 2010). 'Coltrane Changes' are based on the interval cycle of a major 3$^{\text{rd}}$, used to identify three tonal centres within the 12-tone pitch cycle, with free tonal harmonic movement permitted between them. The cycle of tonal centres in *Giant Steps* are B, $E\flat$, and G; with a conventional reading (e.g., Waters, 2010) identifying two descents ($B - G - E\flat$ and $G - E\flat - B$) in bars 1-8, followed by a single prolonged ascent ($E\flat - G - B - E\flat$) in bars 9-15. With the exception of the first tonal centre of B (incorrectly grouped with the following tonal centre of G), IDyOT precisely identifies this tonal structure (Figure 11.4). Again, arguably the $ii - V - I$ shapes defining the tonal centres themselves are over segmented, but as discussed in §11.5.1, this issue is merely of selecting a level of segmentation that is satisfactory to the listener or reader. Interestingly, all of the upper layer transitions between tonal centres are weak, with the exception of the repeated ones, even though Coltrane explicitly permits these otherwise unusual progressions. The upper layer of IDyOT does not have a short term memory (unlike the STM on the surface layer), and can only pick up limited statistical structure within a piece with the LTM+. Similarly, the training data contains only a few other Coltrane lead sheets; with a training set comprising entirely of the composer's pieces similar multiple viewpoint statistical approaches have shown that Coltrane's style can be satisfactorily learned and classified (Hedges et al., 2014).

11.6 Segmenting and Labelling Tonal Harmonic Sequences

The establishment of the IDyOT Tonal Chunker's ability to produce musically meaningful segmentations over a few hand-selected lead sheets (§11.5) prompts further empirical, and a more comprehensive, evaluation. Ordinarily, an empirical evaluation of a segmentation algorithm will compare against a ground truth, reporting accuracy and F-measure scores. However, this is especially problematic for ambiguous tasks, such as tonal harmonic segmentation, with potentially multiple 'correct' interpretations and other interpretations varying in validity. In this case, the concept of a single 'truth' against which evaluations can be made is nonsensical (Pearce et al., 2010b; Wiggins, 2009). Rather, the current study takes multiple plausible segmentations from different sources, looking for agreement between the resulting segmentations.

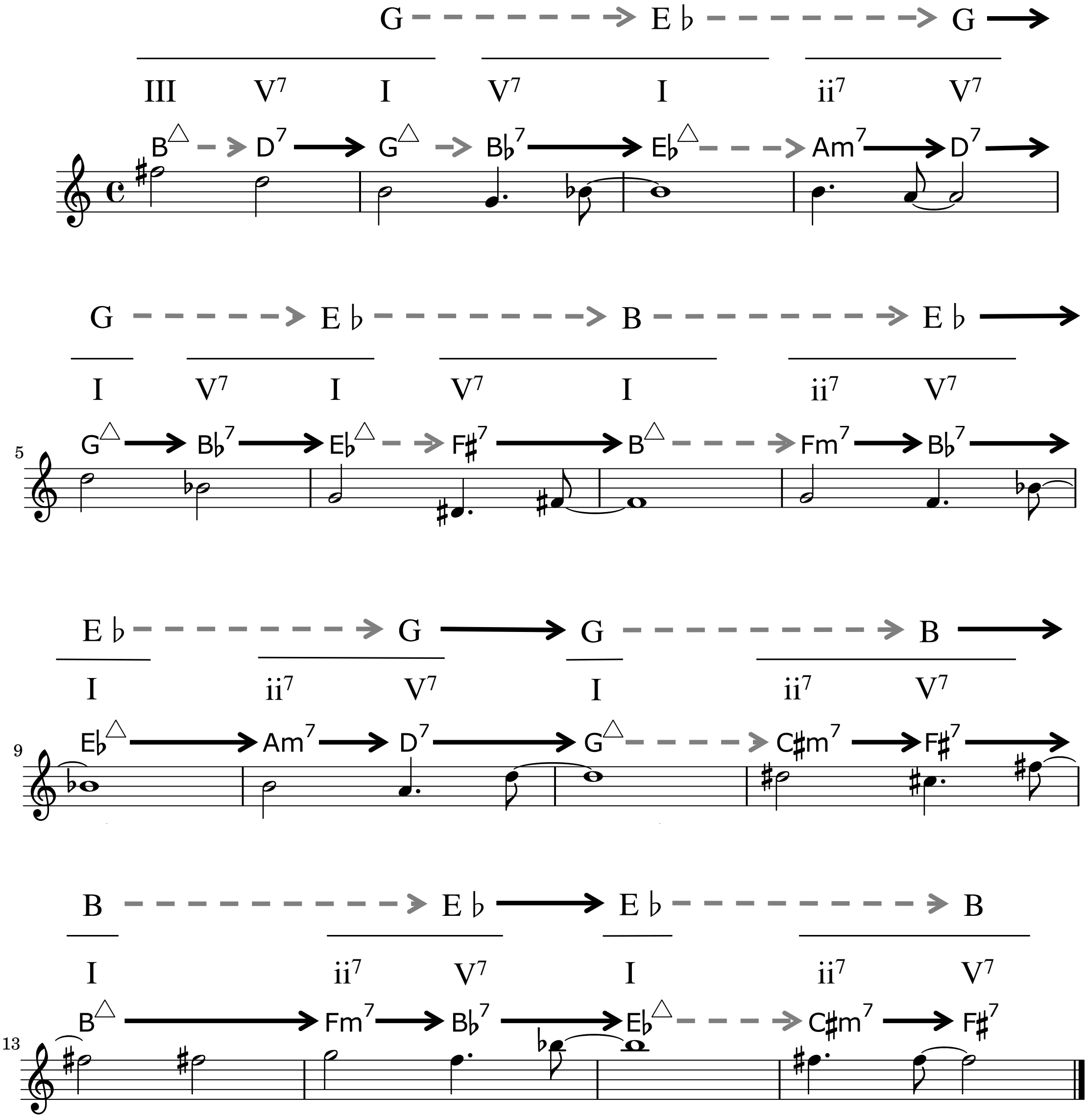


Figure 11.4: IDyOT parsing of *"Giant Steps"* by John Coltrane. Local tonal centres spanning chunks (plain horizontal lines) are shown on the top line, which are used to inform scale degree on the second line. On both levels, high probability transitions are indicated with bold arrows, and low probability with grey.

11.6.1 Experimental Methodology

Agreement between multiple raters (in the current study, segmenters) is given by Fleiss's Kappa (Fleiss, 1971):

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{11.9}$$

where $\bar{P}$ is the observed degree of agreement between raters, and $\bar{P}_e$ the degree of agreement by chance. Therefore, the measure gives an indication of the ratio between the agreement achieved above chance level, with the theoretical maximum agreement above chance. Conger's (1980) exact method for κ is employed so that agreement values between two and three raters can be meaningfully compared.

Three contrasting segmentation methods are chosen for comparison: the IDyOT Tonal Chunker, the rule based system of Pachet (2000), and a human expert. The IDyOT Tonal Chunker (Table 11-G) takes the best performing parameters from §11.4, except the chunking threshold d, which is varied from 0.0 to 4.0. Pachet (2000) describes a non-learned (see §2.3) hierarchical rule based system, matching a pre-defined ontology of patterns, or *shapes* such as *turnarounds*, *two-fives* and *two-five-one's*. The system aims to label chords with tonal centres by minimising modulations whilst fitting to the pre-defined *shapes*.[9] Finally, a human expert[10] applied tonal centres and segment boundaries manually to the corpus, faithfully following established jazz theory practices (Levine, 1989; Levitin, 1994). Each segmentation method applies a tonal centre label (a pitch class from the alphabet of `Root`) and binary value indicating the start of a new chunk to each event, storing them as a tuple. Prior to analysis, none of the segmentation methods considered are any closer to a theoretical ground truth than the others. For reference and reproducibility, the segmentations according to Pachet (2000) and the human expert are given in Appendix F.[11]

Owing to the difficulties associated in obtaining reliable, hand-encoded harmonic analyses, the analysis is conducted on a single fold of the usual 10-fold cross validation. After removing three pieces that could not be interpreted by the Pachet (2000) segmenter, the testing set consisted of 32 lead sheets and 1,328 chords in total. Naturally, the results should be understood in this context; giving potential indications of behaviour and segmentation ability, rather than a thorough, performance-driven assessment of segmentation accuracy.

[9] The analyses were extracted by hand from http://lsdb.flow-machines.com, see Pachet et al. (2013).

[10] The current author. It is worth emphasising that these were made *before* the IDyOT Tonal Chunker was applied to the corpus.

[11] CSV files are available on request from the author.

Table 11-G: IDyOT Tonal Chunker parameters for segmentation.

Parameter	Set of values
Chunk strength measure	H_s
Chunking threshold	$0.0 \leq d \leq 4.0$
Chunking mechanism	*ratio*
Chunk alphabet size	$\vert[\texttt{Root}]\vert$
Tonal centre viewpoint	`LastRootTonal`
Tonal chunk viewpoint	$\texttt{TonalInt}\otimes\texttt{MajType}$
Event bias	8
Chunk bias	8

11.6.2 Hypothesis

It is hoped that a κ value indicating at least some level of agreement between raters will be achieved. For high κ values exact matches must be found between all raters, which is relatively prohibitive considering the ambiguity and representational structure of tonal harmony. As a very broad rule of thumb (Landis & Koch, 1977) suggests $0.4 < \kappa \leq 0.6$ to be 'moderate' agreement, $0.2 < \kappa \leq 0.4$ to be 'fair' agreement, $0.0 < \kappa \leq 0.2$ to be poor, and values less than 0, worse than chance agreement. Given the difficulty of the task, anything above a 'moderate' is unlikely, although performance well above chance is expected. For the IDyOT Tonal Chunker, a threshold of $1.0 \leq d \leq 1.3$ has been shown to produce chunks of approximately the correct length for musical analysis (Table 11-F), which is expected to produce the higher levels of agreement between the segmentation methods.

11.6.3 Results

The three-way agreement between all three segmentation methods is given in Figure 11.5, varying the chunking threshold, d, for the IDyOT Tonal Chunker. Three forms of matching are reported, firstly matching both chunk label (the tonal centre) and the chunk boundary, secondly, the chunk label only, and thirdly the boundary only. When matching both label and boundary, only a 'fair' agreement (best of $\kappa = 0.282$ when $d = 1.0$) is found, suggesting substantial proportions of disagreement between the methods. A higher agreement level of $\kappa = 0.477$ when $d = 0.75$ is found when matching chunk labels only, indicating a substantial level of agreement between segmentation methods on tonal centres. The poor agreement (best of $\kappa = 0.225$ when $d = 1.0$) for boundary only matching can largely be accounted for by the large number of events that do not occur on a boundary, making the chance probability of agreement between segmentation methods relatively high. However, in addition, the Pachet (2000) method does not place boundaries between events with the same tonal centre, whilst the other segmenters are able to if appropriate. This creates a systematic bias against the Pachet (2000) segmenter, and so the fairest comparison between all three models is with chunk labels only, which should be kept in mind for the subsequent two-way comparisons.

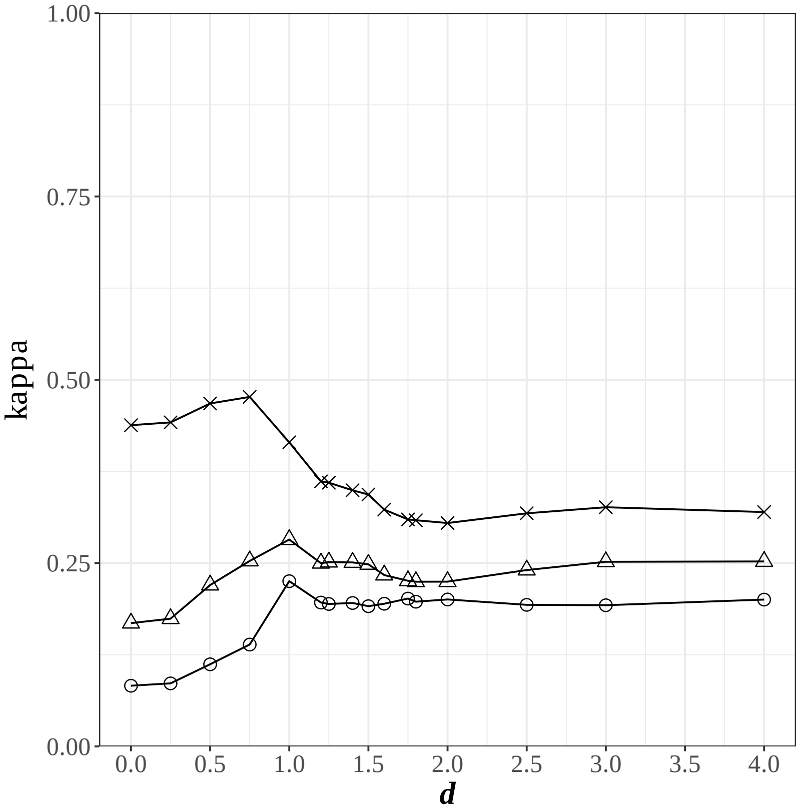


Figure 11.5: Three-way agreement measured by kappa, κ, between human expert, rule-based (Pachet, 2000), and statistical (IDyOT) segmentation methods, varying the chunking threshold, d, for IDyOT. Circles indicate the binary boundary indicator need to be matched, crosses only the chunk label, and triangles both must match.

In order to build a more detailed understanding of the relationships between the three models, the pairwise agreement between models is reported in Table 11-H. For boundary only agreement, the IDyOT Tonal Chunker appears to perform poorly, with only low levels of agreement between the Pachet (2000) method and the segmentations by a human expert. When matching both boundary and chunk label, the Pachet (2000) and IDyOT models agree roughly equally with the expert segmentations, but poorly with each other. The most important finding is the high level of agreement ($\kappa = 0.630$) between IDyOT and the expert segmentations when matching chunk labels only, constituting a 'moderate' to 'substantial' level agreement (Landis & Koch, 1977). As noted above, matching chunk labels only allows all segmentation models to be compared in a completely unbiased manner. It is noteworthy, therefore, that the Pachet (2000) model returns lower levels of agreement ($\kappa = 0.418$) with the expert segmentation than the IDyOT Tonal Chunker.

Table 11-H: Pairwise segmentation agreement as measured by Fleiss's kappa, κ, between three segmentations models.

Segmentation model 1	Segmentation model 2	Boundary ($d = 1.0$)	Chunk label ($d = 0.75$)	Both ($d = 1.0$)
Expert	Pachet (2000)	0.348	0.418	0.352
Expert	IDyOT	0.193	0.630	0.314
Pachet (2000)	IDyOT	0.148	0.383	0.181

The optimal chunking threshold, d, for IDyOT to find its highest levels of agreement is at the bottom end of the hypothesised $1.0 \leq d \leq 1.3$. Indeed, for the chunk label matching task the highest κ is found when $d = 0.75$; a threshold that produces chunk boundaries even with a slight fall in uncertainty. It appears that the IDyOT Tonal Chunker performs optimally as a chunker when on average it produces a chunk length of around two chords or less.

11.7 Discussion and Conclusions

Building on the implementation of IDyOT presented in Chapter 10, the current chapter has presented a modified IDyOT implementation allowing for a wider range of empirical and qualitative testing. The modified approach uses a small amount of domain-specific knowledge relating to identifying tonal centres from cadential figures. It could be argued that this detracts from the strongly bottom-up approach employed elsewhere in this thesis by allowing the system to assume that the tonal centre usually matches, or is related to, the first or last chord of a harmonic chunk. Whilst this concern is valid, it is important to note that the choice of the most sophisticated viewpoint used to identify tonal centres was validated by an information theoretic comparison. This suggests that this component of the cognitive architecture *could* in principle be learned from training data in future work with more advanced implementations of the architecture.

In terms of information theoretic predictive performance, the IDyOT Tonal Chunker suffers from the same shortcomings as its predecessor in Chapter 10; an optimal parametrisation is found when the upper layer prediction is nullified, leaving, essentially, a surface-only IDyOM model. The maximised combination biases suggest that the fundamental issue with the implementation is in integrating the chunk level and surface level predictions coherently. Recalling that multiple viewpoint systems combine predictions in two stages (first individual viewpoint predictions are combined and then the LTM and STM predictions, see Figure 3.1), the manner in which the current IDyOT implementation combines predictions may need to be reconsidered. When predicting the current event, the present strategy of combining the prediction from the upper layer, $p({}_kE^i|{}_kV)$, and the surface context, $p(E^i|e_1^{i-1})$, after the viewpoint and LTM-STM combinations have taken place gives the prediction from the upper layer a substantial degree of importance, equivalent to all of the viewpoints, the LTM, and the STM of the musical surface. This may be problematic as the statistical model of the upper layer is notably less sophisticated than the surface predictions, which use variable order techniques, captures long and short term structure, and takes advantage of the powerful representational properties of viewpoints. A softer approach would integrate the prediction alonside the viewpoint and LTM-STM predictions, enable a subtler probabilistic influence from the upper layer.

The ability for the IDyOT Tonal Chunker to label chunks by tonal centre enables a richer set of evaluation tools involving harmonic analysis to be employed. The specific musicological readings of two well known jazz standards by IDyOT largely match the established harmonic analyses of these pieces. Moreover, a substantial level of agreement is exhibited between IDyOT, a rule-based analysis (Pachet, 2000), and a human expert, when considering chunk labels (but not chunk boundaries), suggesting that all segmentation methods share common ground, if not matching exactly. It is worth highlighting at this stage that IDyOT contains only a very limited amount of domain-specific knowledge. The non-learned knowledge the IDyOT Tonal Chunker possesses is largely at a representational level in the form of viewpoints, which are selected on an information theoretic basis. It is a considerable finding, therefore, that relatively sophisticated harmonic analyses can be conducted by a general architecture that derives its knowledge through statistical learning, performing comparably to a rule-based model designed specifically for that task.

Part IV

Coda

Chapter 12

Conclusions, Reflections, and Future Work

12.1 Thesis Summary

Chapter 1 positions this thesis in the fields of music cognition, computational modelling, and statistical learning. The research follows a strongly bottom-up, statistical account of knowledge acquisition, positing that structurally complex entities such as natural language and music may be learned purely through exposure to (unlabelled) training information and relatively simple learning mechanisms. Established statistical models, such as multiple viewpoint systems (Conklin & Witten, 1995), are powerful models of local structure, but struggle to adequately account for higher order structure, providing the motivation for the current research. The aims of the thesis are established; developing and implementing statistical models with the potential to account for higher order structure. The computational models developed in this thesis serve as high level abstractions of cognitive processes, specifically at the functional level. The focus of the empirical validation of the models is in quantifying their ability to produce information theoretically efficient accounts of unseen data, having processed training data from a similar source.

Chapter 2 aligns the present research with the lineage of Meyer (1956); that emotion and meaning in music arises through structured expectation, which may be systematised (Narmour, 1990), and accounted for in evolutionary terms with statistical learning (Huron, 2006). The body of behavioural research related to statistical learning in music is reviewed, suggesting that statistical learning may account for many cognitive and perceptual processes essential to music; notably expectation, local and non-local harmonic structure, implicit learning, melodic similarity, and grouping. Behavioural and neurophysiological evidence suggests that humans are able to perceive both local and non-local structure in tonal harmony, although non-local dependencies are likely to have an upper bound of 10-12 seconds (Farbood, 2010; Woolhouse et al., 2016), implying that unbounded recursion is not a requirement for computational models of tonal harmonic cognition. A distinction is made between learned and non-learned computational models of tonal harmony, positioning the current research alongside other statistical, machine learning approaches (Paiement, 2008; Ponsford et al., 1999; Raphael & Stoddard, 2004). Application of multiple viewpoint systems are reviewed comprehensively, with the wide range of music informatic, cognitive modelling, and cross-domain tasks motivating their use in the current research. A comprehensive theoretical description of multiple viewpoint systems is given in Chapter 3, providing the theoretical underpinnings of the computational model developed later in the thesis.

The focus of Part II of the thesis is in developing multiple viewpoint systems, specifically in issues related to modelling chord sequences, but not in higher order structure. Chapter 4 defines the viewpoints and corpora used in the current research, additionally defining the musical surface of the thesis to be at the chord symbol level. Chapter 5 deals with the problem of predicting multiple basic attributes in a multiple viewpoint systems, proposing a solution where multiple attributes are merged into a single attribute. Using mean information content as a performance metric, predicting merged attributes is found to be more effective than predicting the same attributes individually when the individual attributes are highly correlated, as is the case with `Root` and `ChordType` (the viewpoints comprising a chord symbol). The results are found to hold for a full multiple viewpoint system, with individual attributes of `Root` and `ChordType` predicted at 3.393 bits/symbol, and merged attributes at 2.963 bits/symbol for a corpus of jazz lead sheets from the *Real Book Vol. 1* (Leonard, 2012; Pachet et al., 2013). The chapter also compares the performance of various smoothing techniques for the domain of chord sequences, broadly confirming the findings for melodic data (Pearce & Wiggins, 2004); that escape method C (Witten-Bell smoothing, Moffat, 1990; Witten & Bell, 1991) is the best escape method, interpolated smoothing (Chen & Goodman, 1999; Jelinek & Mercer, 1980) outperforms backoff smoothing (Kneser & Ney, 1995), and update exclusion (Cleary & Witten, 1984) produces only inconsistent improvements.

The results presented in Chapter 5 suggest that derived viewpoints that abstract heavily from their basic viewpoint (e.g. `FunctionType, MajType, 7-Type, MeeusInt, ChromaDist`) perform poorly when predicting chord sequences. Chapter 6 proposes a modification to the mechanism that assigns probability mass when converting from distributions over derived viewpoints to basic viewpoints. Where an element of a derived viewpoint maps onto multiple elements of a basic viewpoint, the established method (Conklin & Witten, 1995; Pearce, 2005) is to distribute the probability mass from the derived element uniformly between the basic elements. The proposed modification distributes the mass according to the zero-order distribution of elements in the basic viewpoint, so that uncommon basic elements do not acquire disproportionately large amounts of the probability mass. Although the modified method statistically significantly improves predictions for most individual derived viewpoints, notably those derived from `ChordType`, it is found that there is minimal impact on the predictive power for full multiple viewpoint systems. Nevertheless, the weighted method allows for compact multiple viewpoint systems to be selected that permit a small drop in predictive performance for a large reduction in the total number of symbols in the alphabets of the predictive viewpoints.

The viewpoint selection algorithm (§3.5) is studied in Chapter 7, assessing how well the greedy step-wise selection process converges to local minima in the search space of viewpoint systems using mean information content as a heuristic. By starting at random points in the search space, the analysis shows that a small collection of related solutions of similar quality are reached, verifying the empty set start state of the original algorithm (Pearce, 2005, p. 122).

The final study of Part II investigates the performance of absolute (e.g. `Pitch`, `Root`) and relative (e.g. `PitchInt`, `RootInt`) viewpoints, challenging the assumption that relative viewpoints should systematically outperform their absolute counterparts. The original assumption is motivated by the findings that humans primarily use relative cognitive representations of pitch (e.g., Attneave & Olson, 1971; Dowling & Bartlett, 1981; Plantinga & Trainor, 2005), and musicologically, transposed melodies are considered equivalent to one another. Pearce and Wiggins (2012) argue that the former is explained by information theoretic findings of IDyOM that relative viewpoints produce predictive models with a lower mean information content in comparison to absolute viewpoints. However, the present study finds that the comparative performance between relative and absolute viewpoints is inconsistent, and appears to be highly dependant on domain, corpus, and varies when linked with various temporal viewpoints.

Part III of the thesis focuses on developing and constructing computational models capable of capturing higher order structure, underpinned by the multiple viewpoint techniques used in Part II. Chapter 9 presents, as background, the Information Dynamics of Thinking (IDyOT) cognitive architecture of Wiggins and Forth (2015), which builds on the theoretical work presented in Wiggins (2012c). IDyOT implements Baars's (1988) Global Workspace theory, where predictive generators compete for access to an

AI blackboard (Corkill, 1991), with information theoretic heuristics mediating access. The cognitive architecture is intended to be domain independent, and to account for phenomena such as higher order structure, segmentation, ambiguous parsings, and creativity.

An exploratory implementation of IDyOT as a stratified DBN is presented in Chapter 10. An empirical testing of parametrisations of IDyOT using mean information content as a model selection heuristic indicates some fundamental flaws in the implementation; namely that the system becomes optimised as the influence of the upper layer is minimised, and is optimal when the upper layer is nullified. As a result, the best parametrisation of IDyOT that produces meaningful chunks (returning $\bar{h} = 3.392$) is unable to outperform a far simpler IDyOM model (returning $\bar{h} = 3.298$). However, the implementation is able to find musicological meaningful chunks in the form of established cadential patterns ($ii^7 - V^7$, $ii^7 - V^7 - I$, etc.). Further analysis shows that for common chunks (those occurring more than 17 times in memory), IDyOT does outperform IDyOM, suggesting poor generalisation of data at the chunk level as a plausible explanation for poor performance.

Chapter 11 builds on the IDyOT implementation of Chapter 10, developing a model more specific to the tonal harmonic domain. The model is broadly similar in architecture, the fundamental difference being that IDyOT identifies tonal centres of chunks, using them to label the chunk and inform predictions of the current event on the surface layer. Empirical testing of the parameter space reveals the modified implementation suffers from the same drawbacks as the first IDyOT implementation of Chapter 10. However, the modified implementation is capable of producing harmonic analyses that are impressive for a predominantly statistical system, and in a segmentation and labelling task is found to perform comparably, if not preferably, to a rule-based model specifically designed for the task (Pachet, 2000).

12.2 Original Contributions

The current thesis makes a number of meaningful contributions to the fields of multiple viewpoint modelling and computational models of cognition.

The bulk of contributions to multiple viewpoint frameworks are found in Part II of the thesis. Other than the trivial contribution of applying the framework to a relatively unexplored domain (jazz chord sequences), the contributions fall into two broad categories: specific theoretical modifications to the framework, and enhancing general understanding. In the former category, Chapter 5 challenges the long held assumption (Conklin, 1990, p. 69) that basic attributes of multiple viewpoint systems can be considered as statistically independent, and predicted as such. §5.6 shows some basic attributes, notably `Root` and `ChordType`, are highly correlated, and are better predicted by merging the attributes together. Chapter 6 contributes and tests a weighting modification to the inverse viewpoint function, Ψ', showing it to improve the prediction of individual viewpoints that abstract heavily from their surface form. §4.5 introduces the concept of given attributes (applied to temporal attributes in the current research) to the multiple viewpoint framework, allowing certain attributes to be effectively marginalised for the purposes of prediction, but still contribute to sequence matching, and therefore statistical structure when included in linked viewpoints. Finally, §11.2 introduces a class of *relational viewpoints*, viewpoints whose sequences are defined both by the surface events and an additional argument, modelling the relation between the surface events and the referent argument. The additional argument may be defined entirely separately from the surface representation, for example, in the current research `TonalInt` measures the chromatic interval between the current event and a tonal centre as defined by a statistical model. The relational viewpoint class allows the multiple viewpoint framework to distinguish between representations learned dynamically from modelling a dataset, and those provided as the surface input.

In the latter category, Chapter 7 provides a deeper understanding of, and further justification for the viewpoint selection algorithm (§3.5), a greedy step-wise selection algorithm used to automatically construct multiple viewpoint systems. The findings that the algorithm converges on sets of similar solutions from random initialisations in the search space verifies its use in both established and future research. A second analytical contribution to multiple viewpoint systems is found in Chapter 8, showing that the high performance of relative viewpoints (`PitchInt, RootInt`, etc.) is, contrary to common assumption, relatively fragile, and highly dependent on domain, corpus, and correlated temporal information. This finding is likely to extend more generally to most Markov models that use mean information content, or negative log probability, as a performance metric.

The original contributions made to the field of computational models of cognition are mainly found in Part III of the thesis and mostly relate to the IDyOT (Forth et al., 2016; Wiggins, 2012c; Wiggins & Forth, 2015) cognitive architecture. Firstly, the current research provides a minimal, but nevertheless functional in terms of prediction, implementation of IDyOT. Prior to the current research, an implementation that combines predictions from multiple layers had not been implemented. Secondly, a thorough empirical exploration of the parameter space provides a comprehensive understanding of the information theoretic performance and behaviour of the system. Despite failing to outperform a simpler IDyOM model, the implementation of the current research is able to guide future development of the cognitive architecture in an informed manner. In particular, the failed searches identify precisely the components of the implementation that need to be reconsidered in future works, notably the combining of predictions from the surface and upper layers. Thirdly, and more positively, the current research has shown that IDyOT is capable of exhibiting predicted musicological behaviour in terms of storing chunks relating to harmonic structure, informing detailed analyses of lead sheets, and reasonably successfully labelling harmonic chunks and identifying structural segment boundaries. These results add credibility to the cognitive architecture as a whole, suggesting that from basic general principles IDyOT might be capable of performing relatively complex, domain-specific tasks.

12.3 Limitations and Future Works

A number of limitations and provisos should be considered when drawing conclusions from the current research. As with all corpus based machine learning tasks, an obvious restriction is placed on the scope of the research. Strictly, conclusions apply only to the domain and training data used. The current research attempts to minimise this limitation by conducting research over five datasets (see Table 4-E): two datasets of chord sequences of contrasting styles, and three of monophonic melodies in different styles. However, the core work of the thesis concerning higher order structure (Part III) is conducted only over the *Real Book Vol. 1* corpus of jazz standards. Whilst the scientific findings of the present research are limited to this particular jazz corpus, an argument can be made that the corpus itself is a good representation of jazz harmony in general, containing much of the core repertoire and holding a strong didactic status in the jazz community. Therefore, it is not unreasonable to suggest that the findings of this thesis will extrapolate well to jazz harmony, and other related tonal harmonic styles. Additionally, it is anticipated that the learning mechanisms driving the statistical models in IDyOT are general enough to be applied to other domains. At their core, they rely simply on matching and counting symbols, and information theoretic properties that apply broadly to any sequential symbolic data.

An aspect of the current research that is not so easily generalisable is the use of hand constructed viewpoints specific to the domain under study; both in surface and abstracted representations. As an entry point to the model, a finite pool of viewpoints are defined by hand, from which some are selected in an objective manner to produce information theoretically compact models with reference to training and test sets (§3.5).

The use of hand coded viewpoints using expert (musicological) knowledge does prevent an entirely bottom-up approach to modelling, although the objective selection of representations from a possible pool adds credibility to the methodology. The potential for computationally learning the viewpoint representations themselves has been discussed as potential future work by Pearce (2005, pp. 220-221), however, to date this line of research has not been pursued. Pearce (2005) outlines an approach using the GISs of Lewin (1987) to define equivalence classes in a recursive search through the domain of a specified attribute. The current research may be used to develop this proposal within the framework of the IDyOT cognitive architecture. Using IDyOT, a search of equivalence classes between chunks, rather than arbitrary partitions of the domain, could be deployed to learn and define abstracted representations (equivalent to derived viewpoints) above the surface attributes. Representations can take the form of conceptual spaces (Gardenfors, 2000), viewed as a cognitively motivated representational framework encompassing GISs. Interval viewpoint representations that continually overlap events (e.g. `PitchInt`, `RootInt`) may be accounted for with a fully parallel IDyOT implementation that permits parallel segmentations of a single surface layer. Overall, IDyOT has the potential to offer an informed, explanatory account of representation learning.

The current research uses mean information content, $\bar{h}$, (Equation 3.5) as the primary performance metric to compare models. This information theoretic measure is the average number of bits required to encode an event, and, as the number of events tends towards infinity, provides a robust estimate of the cross entropy between a probability distribution predicting an event from a statistical model, and one from the unobservable source. The information content of an event, e^i, given a context, c, (possibly containing both surface and upper layers) is given by Equation 12.1. However, this metric does not adequately measure information flow over non-local dependencies, specifically the information content of events in the indeterminate future. The information content of an event t time steps in the future would be given by Equation 12.3, having calculated the future conditional probability of the event first (Equation 12.2).

$$h(e^i|c) = -\log_2 p(e^i|c) \tag{12.1}$$

$$p(e^{i+t}|c) = \sum_{e_{i+1}^{i+t-1} \in \xi^*} p(e^i|e_{i+t}^{i+t-1}, c) \tag{12.2}$$

$$h(e^{i+t}|c) = -\log_2 p(e^{i+t}|c) \tag{12.3}$$

The mean future information content over a reasonable number of time steps should provide a quantitative distinction between a system capable of learning higher order structure, and a surface-only model. Unfortunately, the measure is computationally expensive to compute, summing over all possible sequences in e of length t, and would likely require a modified Forward algorithm from the HMM literature to compute. The cognitive justification for computing such a measure is that it is hugely advantageous to predict beyond the next event, for example, when judging turn-taking in conversations (see Levinson, 2016; Levinson & Torreira, 2015). Other useful information theoretic measures (such as the predictive information) relating to past, present, and future variables explored by Abdallah and Plumbley (2009) may provide a further platform for computational information theoretic evaluation of cognitive architectures.

The motivation behind using mean information content in cognitive modelling follows the argument that minds are essentially processors of information (Clark, 2013; Dennett, 1996), building structured representations that aim to compress information (Wiggins, 2012c; Wiggins & Forth, 2015).[1] Following this argument, minimising $\bar{h}$ is a heuristic for model selection in cognitive modelling, which in turn coincides with a closer fit with human behavioural data (Pearce & Wiggins, 2006). However, the measure is reliant on the implementation detail of the model, and may be too simplistic an optimisation metric for cognition. It seems highly plausible that humans optimise cognitive processes and representations according to a number heuristics, finding a balance between them. Whilst the current research has used $\bar{h}$ as the primary performance metric in the development phase of implementing preliminary IDyOT architectures, more advanced implementations should employ a broader evaluation approach following that of Desain et al. (1998). Evaluation of cognitive models is rooted in their ability to produce results that agree with human behaviour; for IDyOM, agreement has been found for melodic expectation (Pearce et al., 2010c; Pearce & Wiggins, 2006), segmentation (Pearce et al., 2010b), and memory (Agres et al., 2017). A useful future research contribution would be to replicate these results with equivalent behavioural studies in the domain of tonal harmony. It is only after such a validation that it is possible to make convincing claims about IDyOT's ability to model the cognition of tonal harmony. Exploration and validation into further domains would enable more powerful claims to be made concerning IDyOT and general cognition.

[1] See also Abdallah et al. (2015, p. 159), where structure itself is defined as the representation of an apparently large object with a smaller, compressed description.

Appendix A

Notational Conventions

Sets

$\|S\|$	cardinality of set S
2^S	the power set of S
$S \times S'$	the Cartesian product of S and S'
$\mathbb{Z}$	integers
$\mathbb{Z}^+$	positive integers
$\mathbb{Z}^*$	non-negative integers
$\mathbb{R}$	real numbers

Symbols and Sequences

e	an event
e^i	an event at position i of a sequence
e_i^j	a sequence of events indexed from i to j
ε	the empty sequence
$\mathcal{A}$	an alphabet of symbols
$\mathcal{A}^+$	the *positive closure* of $\mathcal{A}$ (the set of all non-empty sequences composed from elements of $\mathcal{A}$)
$\mathcal{A}^*$	the *Kleene closure*: $\mathcal{A}^+ \bigcup \varepsilon$ (the set of all sequences composed from elements of $\mathcal{A}$, including ε)
$\|\|$	sequence concatenation, e.g. $a\|\|bc \rightarrow abc$

Viewpoint Notation

τ	a typed attribute
τ_i	a typed attribute, indexed i
τ_b	a typed basic attribute
τ_{b_i}	a typed basic attribute, indexed i
$[\tau]$	syntactic domain of τ
$[\tau]'$	syntactic domain of τ that has been seen by the model
$t \in [\tau]$	viewpoint element
$\langle\tau\rangle$	type set of τ
$\llbracket\tau\rrbracket$	semantic domain of τ
$\llbracket\cdot\rrbracket_\tau : [\tau] \rightarrow \llbracket\tau\rrbracket$	semantic interpretation of $[\tau]$
$\llbracket\cdot\rrbracket'_\tau : \llbracket\tau\rrbracket \rightarrow [\tau]$	syntactic interpretation of $\llbracket\tau\rrbracket$
$\Psi_\tau : \xi^* \rightharpoonup [\tau]$	viewpoint function
$\Phi_\tau : \xi^* \rightharpoonup [\tau]^*$	viewpoint matching function
$\Psi'_\tau : \xi^* \times [\tau] \rightarrow 2^{[\tau_b]}$	inverse viewpoint function
τ_l	a typed relational attribute
P	domain of referent symbols
$\Upsilon_{\tau_l} : \xi^* \times \mathrm{P} \rightharpoonup [\tau_l]$	relational viewpoint function
$\Omega_{\tau_l} : \xi^* \times \mathrm{P}^* \rightarrow [\tau_l]^*$	matching function of relational viewpoint
$\Upsilon'_{\tau_l} : \xi^* \times [\tau_l] \times \mathrm{P} \rightarrow 2^{[\tau_b]}$	inverse relational viewpoint function

Probability and Information Theory

$p(e\|c)$ or $p(e^i\|e_1^{i-1})$	probability of event e in sequence given context c
$p(e\|c)$	count of e occurring given context c
$t(c)$	type count given context c
$t_1(c)$	number of types occurring once given context c
$h(e\|c)$	information content of event e given context c
$H(c)$	Shannon entropy of distribution following context c

Appendix B

Chomsky Grammar Hierarchy

An introduction to the Chomsky hierarchy of formal grammars (Chomsky, 1956) is presented in this appendix for convenience. Structural descriptions of language are given by formal grammars, G, characterised by the tuple $\langle V, T, S, P \rangle$. The alphabet of the language consists of finite sets of both *terminal*, T, and *non-terminal*, V, symbols, one of which is the *start* symbol, $S \in V$. A set of *transformation* (or *re-write*) rules, P, defines the allowable transformations between sequences of *terminal* and *non-terminal* symbols. The surface representation, or language, $L(G)$, of a grammar is a subset of all possible sequence of terminal symbols, $L(G) \subset T^*$, re-written with a number (possibly zero) of *re-write rules*, P.

The Chomsky hierarchy describes four types of grammars, with the highest (Type 0) placing least restrictions on the production rules, and the lowest (Type 3) being the most restricted. Type 2 and higher grammars require a stack to parse as the production rules allow for sequences of *non-terminal* symbols embedded in sequences of *terminal* symbols. The hierarchy is a containment hierarchy, so grammar is permitted to use production rules from lower grammars.

Table B-1: Chomsky Hierarchy of Formal Grammars.

Type	Description	Production rules
0	Unrestricted	$\gamma \to \alpha$
1	Context Sensitive	$\alpha A \beta \to \alpha \gamma \beta$
2	Context Free[1]	$A \to Ba$ $A \to aBb$ $A \to \alpha$
3	Finite State[2]	$A \to a$ $A \to aB$

The Chomsky hierarchy is expressed in Table B-1. $a \in T^*$ is a (possibly empty) sequence of *terminal* symbols, $A, B \in V$ are *non-terminal* symbols, $\alpha, \beta \in (T \cup V)^*$ are (possible empty) sequences of *terminal* and *non-terminal* symbols and $\gamma \in (T \cup V)^+$ is a non-empty sequence of *terminal* and *non-terminal* symbols. Type 0 (unrestricted) grammars may re-write any non-empty sequence of symbols (*terminal* or *non-terminal*) with another (possibly empty) sequence. Type 1 (context sensitive) grammars must contain at least one non-terminal symbol on the left hand side, and at least as many symbols (*terminal* or *non-terminal*) on the right. Type 2 (context free) grammars restrict the left-hand side to a single *non-terminal* symbol. Finally, Type 3 (finite state) grammars restrict the left-hand to a single *non-terminal* symbol, re-written as a sequence containing up to one *terminal* symbol.

[1] Type 2 grammars may be expressed more compactly as $A \to \alpha$, but the description given makes explicit how long-term dependencies and embedded structure are arrived at.

[2] This description is of a right-linear grammar (parsed left to right). For a left-linear, grammar replace $A \to aB$ with $A \to Ba$.

Appendix C

Algorithms

Algorithm 2 Algorithm to determine chroma distance from a given root interval, in the style of Cormen, Leiserson, Rivest and Stein (2001).

```
Require: Interval must be an integer $\in [0, 11]$
function CHROMA-DISTANCE($interval$)
    if $interval = 0$ then return $interval$
    end if
    $descending\text{-}fifths \leftarrow 1$
    $descending\text{-}interval \leftarrow interval$
    while $descending\text{-}interval \bmod 7 \neq 0$ do
        $descending\text{-}interval \leftarrow descending\text{-}interval - 12$
        $descending\text{-}fifths \leftarrow descending\text{-}fifths + 1$
    end while
    $ascending\text{-}fifths \leftarrow 1$
    $ascending\text{-}interval \leftarrow interval$
    while $ascending\text{-}interval \bmod 7 \neq 0$ do
        $ascending\text{-}interval \leftarrow ascending\text{-}interval + 12$
        $ascending\text{-}fifths \leftarrow ascending\text{-}fifths + 1$
    end while
    $min\text{-}fifths \leftarrow \min(ascending\text{-}fifths, descending\text{-}fifths)$
    return $min\text{-}fifths$
end function
```

Algorithm 3 Pitch class set categorisation algorithm, in the style of Cormen, Leiserson, Rivest and Stein (2001).

```
Require: pcset = $\{x \mid x \in \mathbb{Z}, x \geq 0, x < 11\}$
function CATEGORISE($pcset$)
    if $4 \in pcset$ then
        if $10 \in pcset$ then
            if $8 \in pcset$ then
                return alt
            else
                return 7
            end if
        else
            if $9 \in pcset$ then
                return 6
            else if $8 \in pcset$ then
                return aug
            else
                return maj
            end if
        end if
    else if $3 \in pcset$ then
        if $10 \in pcset$ then
            if $6 \in pcset$ then
                return halfdim
            else
                return min7
            end if
        else
            if $6 \in pcset$ then
                return dim
            else
                if $8 \in pcset$ then
                    return min♯5
                else
                    return min
                end if
            end if
        end if
    else if $\mid pcset \mid > 0$ then
        if $7 \in pcset$ then
            return sus
        else
            return special
        end if
    else
        return NC
    end if
end function
```

Appendix D

Viewpoint Selection Runs from Random Initialisations

Table D-1: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$
	RootIntFiP⊗FunctionType MeeusInt⊗ChordType⊗PosInBar MeeusInt⊗FunctionType⊗PosInBar MeeusInt⊗7Type ChromaDist⊗ChordType	3.453
Iteration	Viewpoint added/deleted	$\bar{h}$
1	- MeeusInt⊗7Type	3.341
2	- MeeusInt⊗FunctionType⊗PosInBar	3.253
3	+ Root⊗ChordType⊗PosInBar	3.097
4	- MeeusInt⊗ChordType⊗PosInBar	3.083
5	+ RootInt⊗ChordType⊗PosInBar	3.009
6	- ChromaDist⊗ChordType	2.997
7	+ Root⊗ChordType	2.988
8	+ RootInt⊗ChordType	2.975
9	+ RootIntFiP⊗ChordType⊗PosInBar	2.965
10	- RootIntFiP⊗FunctionType⊗	2.962
	Final Viewpoint System	$\bar{h}$
	Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType⊗PosInBar	2.962

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-2: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$

	`RootInt⊗7Type⊗PosInBar`
	`RootIntFiP⊗MajType`
	`Meeusint⊗FunctionType`
	`MeeusIntFiP⊗MajType⊗PosInBar`
	`ChromaDistFiP⊗7Type`
		3.714

Iteration	**Viewpoint added/deleted**	$\bar{h}$
1	- `MeeusIntFiP⊗MajType⊗PosInBar`	3.638
2	- `ChromaDistFiP⊗7Type`	3.567
3	- `Meeusint⊗FunctionType`	3.501
4	+ `Root⊗ChordType⊗PosInBar`	3.179
5	- `RootIntFiP⊗MajType`	3.153
6	+ `RootInt⊗ChordType`	3.044
7	- `RootInt⊗7Type⊗PosInBar`	3.022
8	+ `RootIntFiP⊗ChordType⊗PosInBar`	2.990
9	+ `Root⊗ChordType`	2.982
10	+ `RootInt⊗ChordType⊗PosInBar`	2.962
	Final Viewpoint System	$\bar{h}$

	`Root⊗ChordType`
	`Root⊗ChordType⊗PosInBar`
	`RootInt⊗ChordType`
	`RootInt⊗ChordType⊗PosInBar`
	`RootIntFiP⊗ChordType⊗PosInBar`
		2.962

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict `Root⊗ChordType` in the *Real Book Vol. 1* corpus.

Table D-3: Viewpoint selection from a random initialised set of 5 random viewpoints.

Initial Viewpoint System	$\bar{h}$
Root⊗FunctionType⊗PosInBar Root⊗MajType MeeusInt⊗7Type MeeusInt⊗MajType⊗PosInBar MeeusIntFiP⊗ChordType⊗PosInBar	3.584

Iteration	Viewpoint added/deleted	$\bar{h}$
1	- MeeusInt⊗7Type	3.489
2	- MeeusInt⊗MajType⊗PosInBar	3.403
3	- MeeusIntFiP⊗ChordType⊗PosInBar	3.376
4	+ RootInt⊗ChordType⊗PosInBar	3.085
5	- Root⊗MajType	3.072
6	+ RootIntFiP⊗ChordType	3.000
7	+ Root⊗ChordType	2.984
8	+ RootInt⊗ChordType	2.978
9	+ Root⊗ChordType⊗PosInBar	2.967
10	- Root⊗FunctionType⊗PosInBar	2.967
(11	+ RootIntFiP⊗ChordType⊗PosInBar	2.964)

Final Viewpoint System	$\bar{h}$
Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType	2.967

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-4: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$
	Root⊗FunctionType⊗PosInBar	3.212
	RootInt⊗ChordType⊗PosInBar	
	MeeusInt⊗ChordType	
	MeeusIntFiP⊗ChordType	
	ChromaDistFiP⊗MajType	
Iteration	Viewpoint added/deleted	$\bar{h}$
1	- ChromaDistFiP⊗MajType	3.148
2	- MeeusIntFiP⊗ChordType	3.096
3	- MeeusInt⊗ChordType	3.072
4	+ RootIntFiP⊗ChordType	3.000
5	+ Root⊗ChordType	2.984
6	+ RootInt⊗ChordType	2.978
7	+ Root⊗ChordType⊗PosInBar	2.967
8	- Root⊗FunctionType⊗PosInBar	2.967
(9	+ RootIntFiP⊗ChordType⊗PosInBar	2.964)
	Final Viewpoint System	$\bar{h}$
	Root⊗ChordType	2.967
	Root⊗ChordType⊗PosInBar	
	RootInt⊗ChordType	
	RootInt⊗ChordType⊗PosInBar	
	RootIntFiP⊗ChordType	

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-5: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$
	RootInt⊗FunctionType⊗PosInBar MeeusInt⊗MajType MeeusInt⊗MajType⊗PosInBar ChromaDist⊗FunctionType⊗PosInBar ChromaDistFiP⊗ChordType	3.465
Iteration	Viewpoint added/deleted	$\bar{h}$
1	- MeeusInt⊗MajType	3.359
2	- MeeusInt⊗MajType⊗PosInBar	3.269
3	- ChromaDist⊗FunctionType⊗PosInBar	3.220
4	+ Root⊗ChordType⊗PosInBar	3.068
5	+ RootInt⊗ChordType	3.010
6	+ RootIntFiP⊗ChordType⊗PosInBar	3.000
7	- ChromaDistFiP⊗ChordType	2.983
8	+ Root⊗ChordType	2.969
(9	+ RootInt⊗ChordType⊗PosInBar	2.965)
	Final Viewpoint System	$\bar{h}$
	Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗FunctionType⊗PosInBar RootIntFiP⊗ChordType⊗PosInBar	2.969

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-6: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$
	RootInt⊖FiB⊗7Type⊗ RootInt⊖FiB⊗7Type⊗PosInBar MeeusInt⊗ChordType⊗PosInBar ChromaDistFiP⊗FunctionType⊗PosInBar ChromaDistFiP⊗ChordType⊗PosInBar	3.476
Iteration	Viewpoint added/deleted	$\bar{h}$
1	- RootInt⊖FiB⊗7Type⊗PosInBar	3.452
2	- ChromaDistFiP⊗FunctionType⊗PosInBar	3.436
3	+ RootInt⊗ChordType	3.181
4	- RootInt⊖FiB⊗7Type⊗	3.146
5	- MeeusInt⊗ChordType⊗PosInBar	3.127
6	+ Root⊗ChordType⊗PosInBar	3.021
7	+ RootInt⊗ChordType⊗PosInBar	2.999
8	+ Root⊗ChordType	2.977
(9	+ RootIntFiP⊗ChordType	2.973)
	Final Viewpoint System	$\bar{h}$
	Root⊗ChordType Root⊗ChordType⊗PosInBar RootInt⊗ChordType RootInt⊗ChordType⊗PosInBar ChromaDistFiP⊗ChordType⊗PosInBar	2.977

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-7: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$

	`RootInt⊗ChordType⊗PosInBar`
	`MeeusIntFiP⊗FunctionType`
	`MeeusIntFiP⊗7Type`
	`MeeusIntFiP⊗ChordType⊗PosInBar`
	`ChromaDistFiP⊗ChordType⊗PosInBar`
		3.400

Iteration	Viewpoint added/deleted	$\bar{h}$
1	- `MeeusIntFiP⊗7Type⊗`	3.290
2	- `MeeusIntFiP⊗FunctionType`	3.203
3	- `MeeusIntFiP⊗ChordType⊗PosInBar`	3.136
4	+ `Root⊗ChordType`	3.008
5	+ `RootInt⊗ChordType`	2.998
6	+ `Root⊗ChordType⊗PosInBar`	2.977
(7	+ `RootIntFiP⊗ChordType`	2.973)
	Final Viewpoint System	$\bar{h}$

	`Root⊗ChordType`
	`Root⊗ChordType⊗PosInBar`
	`RootInt⊗ChordType`
	`RootInt⊗ChordType⊗PosInBar`
	`ChromaDistFiP⊗ChordType⊗PosInBar`
		2.977

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict `Root⊗ChordType` in the *Real Book Vol. 1* corpus.

Table D-8: Viewpoint selection from a random initialised set of 5 random viewpoints.

Initial Viewpoint System	$\bar{h}$
Rootint⊖FiB⊗7Type⊗PosInBar	3.344
MeeusInt⊗ChordType⊗PosInBar	
ChromaDist⊗ChordType	
ChromaDist⊗ChordType⊗PosInBar	
ChromaDistFiP⊗FunctionType	

Iteration	Viewpoint added/deleted	$\bar{h}$
1	- Rootint⊖FiB⊗7Type⊗PosInBar	3.335
2	+ Root⊗ChordType⊗PosInBar	3.164
3	- ChromaDistFiP⊗FunctionType	3.154
4	- ChromaDist⊗ChordType⊗PosInBar	3.136
5	- MeeusInt⊗ChordType⊗PosInBar	3.122
6	+ RootInt⊗ChordType⊗PosInBar	3.038
7	+ RootIntFiP⊗ChordType	2.992
8	- ChromaDist⊗ChordType	2.978
(9	+ Root⊗ChordType	2.976)

Final Viewpoint System	$\bar{h}$
Root⊗ChordType⊗PosInBar	2.978
RootInt⊗ChordType⊗PosInBar	
RootIntFiP⊗ChordType	

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-9: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$
	RootInt⊗FunctionType	3.516
	RootInt⊗7Type⊗PosInBar	
	MeeusInt⊗ChordType⊗PosInBar	
	MeeusIntFiP⊗7Type⊗PosInBar	
	MeeusIntFiP⊗MajType⊗PosInBar	
Iteration	Viewpoint added/deleted	$\bar{h}$
1	- MeeusIntFiP⊗7Type⊗PosInBar	3.41163
2	- MeeusIntFiP⊗MajType⊗PosInBar	3.325
3	+ Root⊗ChordType⊗PosInBar	3.121
4	- RootInt⊗7Type⊗PosInBar	3.098
5	- MeeusInt⊗ChordType⊗PosInBar	3.077
6	+ RootInt⊗ChordType⊗PosInBar	3.023
7	+ RootIntFiP⊗ChordType	2.979
8	- RootInt⊗FunctionType	2.978
(9	+ Root⊗ChordType	2.976)
	Final Viewpoint System	$\bar{h}$
	Root⊗ChordType⊗PosInBar	2.978
	RootInt⊗ChordType⊗PosInBar	
	RootIntFiP⊗ChordType	

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Table D-10: Viewpoint selection from a random initialised set of 5 random viewpoints.

	Initial Viewpoint System	$\bar{h}$
	Root⊗ChordType RootIntFiP⊗7Type⊗PosInBar MeeusInt⊗MajType ChromaDist⊗7Type ChromaDistFiP⊗FunctionType⊗PosInBar	3.425
Iteration	Viewpoint added/deleted	$\bar{h}$
1	- MeeusInt⊗MajType	3.339
2	- ChromaDist⊗7Type	3.282
3	- ChromaDistFiP⊗FunctionType⊗PosInBar	3.227
4	+ RootInt⊗ChordType⊗PosInBar	3.024
5	- RootIntFiP⊗7Type⊗PosInBar	3.013
6	+ RootIntFiP⊗ChordType⊗PosInBar	2.979
(7	+ RootInt⊗ChordType	2.979
	Final Viewpoint System	$\bar{h}$
	Root⊗ChordType RootInt⊗ChordType⊗PosInBar RootIntFiP⊗ChordType⊗PosInBar	2.979

Note. A STMC*IU-LTMC*I model, weighting Ψ', and using bias weights of $b = 2$ and $b = 1$ for LTM(+)-STM and viewpoint combination respectively, is used to predict Root⊗ChordType in the *Real Book Vol. 1* corpus.

Appendix E

Supplementary IDyOT Parametrisation Results

Table E-1: Performance of IDyOT using information content, $h({}_ke^i|{}_ke_1^{i-1}, u, {}_kV)$, as a chunk strength measure.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.0	1.988	69.31%	4.068
0.5	2.290	60.06%	4.080
1.0	2.694	51.94%	4.084
2.0	4.240	49.42%	4.071
3.0	7.769	54.95%	4.074
4.0	14.841	96.60%	4.068
5.0	25.933	100.00%	4.079
6.0	35.342	100.00%	4.082
7.0	40.962	100.00%	4.083
8.0	42.097	100.00%	4.084

Table E-2: Performance of IDyOT using entropy, $H({}_kE^i|{}_ke_1^{i-1}, u, {}_kV)$, as a chunk strength measure.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	2.095	59.06%	4.160
0.25	2.743	41.61%	4.158
0.50	4.057	39.72%	4.120
0.75	5.141	41.57%	4.107
1.00	5.879	47.29%	4.095
1.50	7.296	57.66%	4.090
2.00	9.528	67.97%	4.086
2.50	12.749	84.21%	4.082
3.00	17.900	99.26%	4.076
3.50	23.671	100.00%	4.076
4.00	30.213	100.00%	4.077

Table E-3: Performance of IDyOT using surface information content, $h_s(e^i|e_1^{i-1})$, as a chunk strength measure.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.0	2.128	60.00%	4.084
0.5	2.504	50.68%	4.087
1.0	2.960	43.77%	4.082
2.0	4.135	39.90%	4.071
3.0	5.849	45.71%	4.066
4.0	8.241	56.98%	4.069
5.0	12.128	75.67%	4.063
6.0	18.266	100.00%	4.065
7.0	26.157	100.00%	4.079
8.0	35.424	100.00%	4.080

Table E-4: Performance of IDyOT using surface entropy, $H_s(E^i|e_1^{i-1})$, as a chunk strength measure.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	2.139	49.47%	3.706
0.20	2.461	42.50%	3.761
0.25	2.555	41.51%	3.773
0.40	2.842	38.68%	3.817
0.50	3.061	37.14%	3.835
0.60	3.287	36.80%	3.856
0.75	3.653	35.64%	3.881
0.80	3.768	36.13%	3.886
1.00	4.343	38.08%	3.914
1.50	5.925	45.15%	3.958
2.00	8.088	56.95%	3.992
2.50	11.282	74.64%	4.014
3.00	16.867	99.96%	4.038
3.50	24.832	100.00%	4.06
4.00	35.178	100.00%	4.077

Table E-5: Performance of IDyOT with a chunking mechanism that signals chunk boundaries when the *absolute* value of surface entropy exceeds a threshold.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	1.000	100.00%	3.298
1.00	1.005	100.00%	3.298
2.00	1.075	100.00%	3.308
2.50	1.136	100.00%	3.324
3.00	1.210	97.66%	3.354
3.50	1.302	91.39%	3.397
4.00	1.461	81.90%	3.471
4.50	1.777	68.46%	3.594
5.00	2.571	48.83%	3.766
5.50	4.786	41.60%	3.939
6.00	13.593	88.55%	4.041
6.50	38.669	100.00%	4.077
7.00	43.670	100.00%	4.083
8.00	43.670	100.00%	4.084

Table E-6: Performance of IDyOT with a chunking mechanism that signals chunk boundaries when the *delta* between adjacent surface entropy values exceeds a threshold.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	2.139	49.47%	3.706
0.20	2.461	42.50%	3.761
0.25	2.555	41.51%	3.773
0.40	2.842	38.68%	3.817
0.50	3.061	37.14%	3.835
0.60	3.287	36.80%	3.856
0.75	3.653	35.64%	3.881
0.80	3.768	36.13%	3.886
1.00	4.343	38.08%	3.914
1.50	5.925	45.15%	3.958
2.00	8.088	56.95%	3.992
2.50	11.282	74.64%	4.014
3.00	16.867	99.96%	4.038
3.50	24.832	100.00%	4.060
4.00	35.178	100.00%	4.077

Table E-7: Performance of IDyOT with a chunking mechanism that signals chunk boundaries when the *ratio* between adjacent surface entropy values exceeds a threshold.

Chunking threshold	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	1.000	100.00%	3.298
0.25	1.010	100.00%	3.298
0.50	1.090	100.00%	3.313
0.75	1.224	95.66%	3.376
1.00	2.139	49.47%	3.706
1.25	4.311	39.01%	3.917
1.50	6.311	48.63%	3.964
1.75	8.032	61.91%	3.987
2.00	10.098	68.59%	4.008
4.00	32.752	100.00%	4.071
8.00	42.688	100.00%	4.083

Table E-8: Performance of IDyOT with a chunking mechanism that signals chunk boundaries when the surface entropy is d standard deviations above a mean calculated with a *uniform window*.

Chunking threshold (d)	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	3.156	37.82%	3.839
0.10	3.399	37.05%	3.856
0.20	3.704	36.55%	3.878
0.25	3.877	36.39%	3.888
0.30	4.110	35.93%	3.903
0.40	4.640	36.63%	3.929
0.50	5.357	38.55%	3.952
0.60	6.259	41.73%	3.974
0.70	7.629	48.25%	3.998
0.75	8.364	51.99%	4.007
0.80	9.318	58.50%	4.017
0.90	11.781	74.42%	4.032
1.00	15.076	93.93%	4.046
1.50	34.856	100.00%	4.076
2.00	42.331	100.00%	4.080

Table E-9: Performance of IDyOT with a chunking mechanism that signals chunk boundaries when the surface entropy is d standard deviations above a weighted mean calculated with a *triangular window*.

Chunking threshold (d)	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	2.751	38.59%	3.802
0.10	2.889	38.16%	3.815
0.20	3.065	37.12%	3.833
0.25	3.168	37.17%	3.840
0.30	3.283	36.30%	3.852
0.40	3.552	36.04%	3.867
0.50	3.893	35.85%	3.888
0.60	4.380	35.84%	3.915
0.70	5.030	36.87%	3.939
0.75	5.370	37.95%	3.950
0.80	5.803	39.80%	3.962
0.90	6.933	44.64%	3.986
1.00	8.452	53.31%	4.007
1.50	21.587	100.00%	4.060
2.00	33.181	100.00%	4.076

Table E-10: Performance of IDyOT with a chunking mechanism that signals chunk boundaries when the surface entropy is d standard deviations above a weighted mean calculated with an *exponential window.*

Chunking threshold (d)	Mean chunk length	Chunk coverage	$\bar{h}$
0.00	2.364	44.21%	3.741
0.10	2.421	42.92%	3.750
0.20	2.480	42.71%	3.757
0.25	2.520	42.01%	3.764
0.30	2.561	41.57%	3.767
0.40	2.666	40.36%	3.778
0.50	2.784	39.32%	3.793
0.60	2.940	38.54%	3.808
0.70	3.156	36.80%	3.830
0.75	3.292	35.99%	3.840
0.80	3.444	35.70%	3.849
0.90	3.793	35.11%	3.873
1.00	4.224	35.38%	3.897
1.50	7.359	48.02%	3.985
2.00	11.636	73.36%	4.024

Table E-11: Each state searched by a greedy hill climbing algorithm selecting selecting the locally optimal chunk and event biases in IDyOT, initialised at $b_c = 1$, $b_e = 1$.

Iteration	Chunk bias (b_c)	Event bias (b_e)	$\bar{h}$
start	**1**	**1**	**3.816**
1	0	0	4.005
1	0	1	3.813
1	**0**	**2**	**3.699**
1	1	0	4.007
1	1	2	3.702
1	2	0	4.009
1	2	1	3.819
1	2	2	3.705
2	**0**	**3**	**3.628**
2	1	3	3.631
3	**0**	**4**	**3.581**
3	1	4	3.583
4	**0**	**5**	**3.546**
4	1	5	3.549
5	**0**	**6**	**3.521**
5	1	6	3.524
6	**0**	**7**	**3.502**
6	1	7	3.504
7	**0**	**8**	**3.487**
7	1	8	3.489

Note. Selected states are indicated in **bold**.

Table E-12: Performance of IDyOT using event biases of $8 > b_e \leq 16$, with $b_c = 0$.

Chunk bias (b_c)	Event bias (b_e)	$\bar{h}$
0	9	3.475
0	10	3.465
0	11	3.457
0	12	3.451
0	13	3.445
0	14	3.441
0	15	3.437
0	16	3.433

Table E-13: Selected chunk and event biases at each iteration of a greedy hill climbing algorithm, initialised at $b_c = 5$, $b_e = 5$.

Iteration	Chunk bias (b_c)	Event bias (b_e)	$\bar{h}$
start	5	5	3.558
1	4	6	3.531
2	3	7	3.510
3	2	8	3.492
4	1	8	3.489
5	0	8	3.487

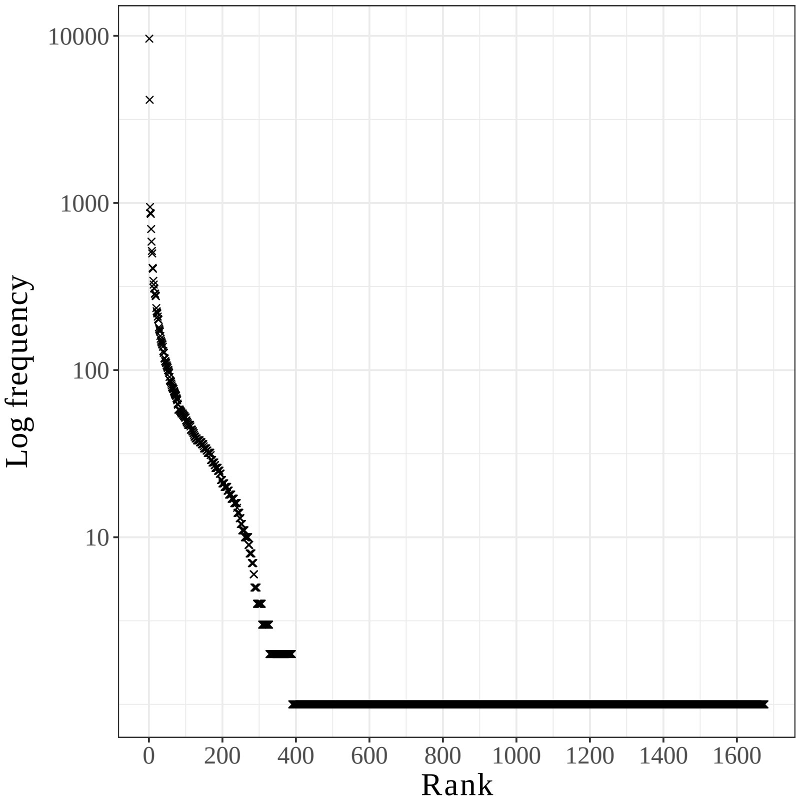


Figure E-1: Log frequency of chunks stored by IDyOT by rank for a complete training and test procedure summed over all folds of a 10-fold cross validation.

Appendix F

Hand and Rule-based Segmentations of Jazz Lead Sheets

Alice in Wonderland by Fain/Hilliard					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	D/m7	1	C	1	C
2	G/7	0	C	0	C
3	C/M	0	C	0	C
4	F/M	1	A	0	C
5	B/halfdim	0	A	1	A
6	E/7	0	A	0	A
7	A/m7	0	A	0	A
8	E♭/7	1	C	1	E♭
9	D/m7	0	C	1	C
10	G/7	0	C	0	C
11	E/m7	1	A	0	C
12	A/m7	0	A	0	C
13	D/m7	1	C	0	C
14	G/7	0	C	0	C
15	E/7	1	A	1	E
16	A/7	1	D	0	E
17	D/m7	1	C	1	C
18	G/7	0	C	0	C
19	D/m7	1	C	1	C
20	G/7	0	C	0	C
21	C/M	0	C	0	C
22	F/M	1	A	0	C

Continued on next page

Alice in Wonderland by Fain/Hilliard					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
23	B/halfdim	0	A	1	A
24	E/7	0	A	0	A
25	A/m7	0	A	0	A
26	E♭/7	1	C	1	E♭
27	D/m7	0	C	1	C
28	G/7	0	C	0	C
29	E/m7	1	A	0	C
30	A/m7	0	A	0	C
31	D/m7	1	C	0	C
32	G/7	0	C	0	C
33	C/M	0	C	0	C
34	A/m7	1	G	0	C
35	D/7	0	G	0	C
36	G/7	1	C	0	C
37	E/m7	1	A	1	A♭
38	A/m7	0	A	0	A♭
39	D/m7	1	C	0	A♭
40	G/7	0	C	1	E♭
41	C/M	0	C	1	C
42	F/M	1	F	0	C
43	F♯/m7	1	E	0	C
44	B/7	0	E	0	C
45	E/m7	0	E	0	C
46	A/7	1	D	0	C
47	D/m7	0	D	0	C
48	A/7	1	D	0	C
49	D/m7	0	D	0	C
50	A/7	1	D	0	C
51	D/m7	0	D	0	C
52	A♭/7	1	D	0	C
53	G/7	0	D	0	C
54	D/m7	1	C	1	A♭
55	G/7	0	C	0	A♭
56	C/M	0	C	0	A♭
57	F/M	1	A	1	E♭
58	B/halfdim	0	A	1	C
59	E/7	0	A	0	C
60	A/m7	0	A	0	C
61	E♭/7	1	C	0	C
62	D/m7	0	C	0	C
63	G/7	0	C	1	G
64	E/m7	1	A	0	G

Continued on next page

Alice in Wonderland by Fain/Hilliard

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
65	A/m7	0	A	0	G
66	D/m7	1	C	0	G
67	G/7	0	C	0	G
68	C/M	0	C	0	G
69	C/M	0	C	0	C

Au Privave by Charlie Parker

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	F/M	1	F	1	F
2	G/m7	1	F	0	F
3	C/7	0	F	0	F
4	F/M	0	F	0	F
5	G/m7	1	B♭	0	F
6	C/m7	0	B♭	1	B♭
7	F/alt	0	B♭	0	B♭
8	B♭/7	0	B♭	1	B♭
9	B♭/m7	1	A♭	1	B♭
10	E♭/7	0	A♭	0	B♭
11	F/M	1	F	0	B♭
12	G/m7	0	F	1	G
13	A/m7	1	G	0	G
14	D/7	0	G	0	G
15	G/m7	0	G	0	G
16	G/m7	1	F	0	G
17	C/7	0	F	0	G
18	F/M	0	F	1	F
19	D/7	1	G	1	G
20	G/m7	0	C	0	G
21	C/7	1	F	1	F
22	F/M	0	F	0	F

Beautiful Love by Victor Young

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	E/halfdim	1	D	1	D
2	A/7	0	D	0	D
3	D/m	0	D	0	D
4	D/m	0	D	0	D
5	G/m7	1	F	1	F

Continued on next page

Beautiful Love by Victor Young

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
6	C/7	0	F	0	F
7	F/M	0	F	0	F
8	E/halfdim	1	D	1	D
9	A/7	0	D	0	D
10	D/m7	0	D	0	D
11	G/m7	1	F	0	D
12	B♭/7	0	F	0	D
13	E/halfdim	1	D	0	D
14	A/7	0	D	0	D
15	D/m7	0	D	0	D
16	G/7	1	D	1	F
17	E/halfdim	1	D	1	D
18	A/7	0	D	0	D
19	E/halfdim	1	D	0	D
20	A/7	0	D	0	D
21	D/m	0	D	0	D
22	D/m	0	D	0	D
23	G/m7	1	F	1	F
24	C/7	0	F	0	F
25	F/M	0	F	0	F
26	E/halfdim	1	D	1	D
27	A/7	0	D	0	D
28	D/m7	0	D	0	D
29	G/m7	1	F	0	D
30	B♭/7	0	F	0	D
31	E/halfdim	1	D	0	D
32	A/7	0	D	0	D
33	D/m7	0	C	0	D
34	B♭/7	1	D	0	D
35	A/7	0	D	0	D
36	D/m	0	D	0	D
37	D/m	0	D	1	F

Beneath it All by Gary Anderson

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	A♭/M	1	E♭	1	E♭
2	A♭/M	0	E♭	0	E♭
3	G/m7	0	E♭	0	E♭
4	G/m7	0	E♭	0	E♭
5	A♭/M	1	B♭	0	E♭
6	A♭/M	0	B♭	0	E♭

Continued on next page

Beneath it All by Gary Anderson

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
7	C/dim	0	B♭	1	G
8	C/dim	0	B♭	0	G
9	E♭/M	1	D	0	G
10	E♭/M	0	D	0	G
11	D/M	0	D	0	G
12	D/M	0	D	0	G
13	B♭/M	1	D	1	D
14	B♭/M	0	D	0	D
15	A/7	0	D	0	D
16	A/7	0	D	0	D
17	B♭/M	1	B♭	0	D
18	B♭/M	0	B♭	0	D
19	C/M	1	C	1	C
20	C/M	0	C	0	C
21	D♭/M	1	C♯	1	C♯
22	D♭/M	0	C♯	0	C♯
23	D♭/M	0	C♯	0	C♯
24	D♭/M	0	C♯	0	C♯

Big Nick by John Coltrane

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	G/M	1	G	1	G
2	E/7	1	A	1	E
3	A/m7	1	G	1	G
4	D/7	0	G	0	G
5	G/M	0	G	0	G
6	E/7	1	A	1	A
7	A/m7	1	G	0	A
8	D/7	0	G	0	A
9	G/M	0	G	1	C
10	B/dim	1	C	0	C
11	C/7	0	C	0	C
12	D♭/dim	1	G	1	B
13	G/M	0	G	0	B
14	E/7	1	A	1	E
15	A/m7	1	G	1	G
16	D/7	0	G	0	G
17	G/M	0	G	0	G
18	E/7	1	A	1	E
19	A/m7	1	G	1	G
20	D/7	0	G	0	G

Continued on next page

Big Nick by John Coltrane

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
21	G/M	0	G	0	G
22	E/7	1	A	1	A
23	A/m7	1	G	0	A
24	D/7	0	G	0	A
25	G/M	0	G	1	C
26	B/dim	1	C	0	C
27	C/7	0	C	0	C
28	D♭/dim	1	G	1	B
29	G/M	0	G	0	B
30	E/7	1	A	1	E
31	A/m7	1	G	1	G
32	D/7	0	G	1	E
33	G/M	0	G	1	G
34	E/7	1	A	0	G
35	A/m7	1	G	0	G
36	D/7	0	G	0	G
37	G/M	0	G	0	G

Blue in Green by Miles Davis

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	B♭/M	1	D	1	D
2	A/7	0	D	0	D
3	D/m7	0	D	0	D
4	D♭/7	1	B♭	1	C♯
5	C/m7	0	B♭	1	B♭
6	F/7	0	B♭	0	B♭
7	B♭/M	0	B♭	0	B♭
8	A/alt	1	D	1	A
9	D/m	0	D	1	A
10	E/7	1	A	0	A
11	A/m7	0	A	0	A
12	D/m7	1	D	1	D
13	B♭/M	0	D	0	D
14	A/7	0	D	0	D
15	D/m	0	D	0	D

Conception by George Shearing

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
1	E♭/halfdim	1	C♯	1	B♭
2	A♭/7	0	C♯	0	B♭
3	D♭/M	0	C♯	1	F♯
4	B/m7	1	A	0	F♯
5	A/M	0	A	1	C♯
6	A♭/M	1	A♭	0	C♯
7	A♭/m7	0	A♭	1	F♯
8	A♭/m7	1	F♯	0	F♯
9	D♭/7	0	F♯	0	F♯
10	F♯/7	1	B♭	0	F♯
11	F/7	0	B♭	1	F
12	B♭/7	1	D	0	F
13	A/7	0	D	1	C♯
14	A♭/7	1	G	0	C♯
15	G/7	0	G	1	G
16	F♯/m7	1	E	1	E
17	B/7	0	E	0	E
18	E/M	0	E	0	E
19	A/M	1	A	0	E
20	E♭/m7	1	C♯	1	C♯
21	A♭/7	0	C♯	0	C♯
22	D♭/M	0	C♯	0	C♯
23	E♭/halfdim	1	C♯	1	C♯
24	A♭/7	0	C♯	0	C♯
25	D♭/M	0	C♯	1	F♯
26	B/m7	1	A	0	F♯
27	A/M	0	A	1	C♯
28	A♭/M	1	A♭	0	C♯
29	A♭/m7	0	A♭	1	F♯
30	A♭/m7	1	F♯	0	F♯
31	D♭/7	0	F♯	0	F♯
32	F♯/7	1	B♭	0	F♯
33	F/7	0	B♭	1	F
34	B♭/7	1	D	0	F
35	A/7	0	D	1	C♯
36	A♭/7	1	G	0	C♯
37	G/7	0	G	1	G
38	F♯/m7	1	E	1	E
39	B/7	0	E	0	E
40	E/M	0	E	0	E
41	A/M	1	A	0	E
42	E♭/m7	1	C♯	1	E♭

Continued on next page

Conception by George Shearing					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
43	A♭/7	0	C♯	0	E♭
44	D♭/M	0	C♯	1	C♯
45	F♯/m7	1	E	0	C♯
46	B/alt	0	E	1	F♯
47	E/M	0	E	1	C♯
48	F♯/m7	1	E	0	C♯
49	A♭/m7	1	F♯	1	F♯
50	D♭/7	0	F♯	0	F♯
51	G/m7	1	F	0	F♯
52	C/7	0	F	1	F
53	F♯/m7	1	E	0	F
54	B/7	0	E	1	C♯
55	E/m7	1	E♭	1	G
56	A/7	0	E♭	1	E
57	E♭/halfdim	1	C♯	0	E
58	A♭/7	0	C♯	1	E♭
59	D♭/M	0	C♯	1	C♯
60	B/m7	1	A	0	C♯
61	A/M	0	A	0	C♯
62	A♭/M	1	A♭	0	C♯
63	A♭/m7	0	A♭	0	C♯
64	A♭/m7	1	F♯	1	F♯
65	D♭/7	0	F♯	0	F♯
66	F♯/7	1	B♭	1	C♯
67	F/7	0	B♭	0	C♯
68	B♭/7	1	D	1	F♯
69	A/7	0	D	0	F♯
70	A♭/7	1	G	0	F♯
71	G/7	0	G	1	F
72	F♯/m7	1	E	0	F
73	B/7	0	E	1	C♯
74	E/M	0	E	0	C♯
75	A/M	1	A	1	G
76	E♭/m7	1	C♯	1	E
77	A♭/7	0	C♯	0	E
78	D♭/M	0	C♯	0	E

Crescent by John Coltrane					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
	chords	hand.boundaries	hand.labels	pachet.boundaries	pachet.labels
1	G/sus	1	C	1	G

Continued on next page

Crescent by John Coltrane					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
2	G/sus	0	C	0	G
3	G/sus	0	C	0	G
4	G/sus	0	C	0	G
5	G/sus	0	C	0	G
6	G/sus	0	C	0	G
7	G/sus	0	C	0	G
8	D/sus	1	G	1	D
9	D/sus	0	G	0	D
10	D/sus	0	G	0	D
11	D/sus	0	G	0	D
12	D/sus	0	G	0	D
13	E/halfdim	1	D	0	D
14	A/7	0	D	0	D
15	D/m7	0	D	0	D
16	G/sus	1	C	1	C
17	G/7	0	C	0	C
18	C/m7	0	C	0	C
19	B♭/sus	1	E♭	1	E♭
20	B♭/7	0	E♭	0	E♭
21	E♭/m7	0	E♭	0	E♭
22	E/m7	1	E	1	E
23	A/alt	0	E	1	D
24	D/sus	0	E	0	D
25	E/halfdim	1	E	0	D
26	A/7	0	E	0	D
27	D/m7	0	E	0	D
28	G/sus	1	C	1	G
29	G/7	0	C	1	G
30	C/m7	0	C	1	B♭
31	B♭/sus	1	E♭	0	B♭
32	B♭/7	0	E♭	0	B♭
33	E♭/M	0	E♭	0	B♭
34	A/alt	1	C	1	F♯
35	D/halfdim	0	C	1	C
36	G/sus	0	C	0	C
37	C/m7	0	C	0	C
38	C/m7	1	C	0	C
39	C/m7	0	C	0	C
40	C/m7	0	C	0	C
41	C/m7	0	C	0	C
42	C/m7	0	C	0	C
43	C/m7	0	C	0	C

Continued on next page

Crescent by John Coltrane					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
44	C/m7	0	C	0	C
45	C/m7	0	C	0	C
46	B♭/sus	1	E♭	1	B♭
47	B♭/sus	0	E♭	0	B♭
48	E♭/m7	0	E♭	0	B♭
49	E♭/m7	0	E♭	0	B♭
50	E/halfdim	1	D	1	D
51	A/alt	0	D	0	D
52	D/m7	0	D	0	D
53	D/m7	1	C	0	D
54	G/sus	0	C	1	C
55	G/sus	0	C	0	C
56	C/m7	0	C	0	C
57	C/m7	0	C	0	C

Day Waves by Chick Corea					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	D/m7	1	B♭	1	B♭
2	B♭/M	0	B♭	0	B♭
3	G/m	1	G	0	B♭
4	E/m7	1	C	1	C
5	F/M	0	C	0	C
6	G/7	0	C	0	C
7	A/m	1	G	1	A
8	D/7	0	G	0	A
9	E/7	1	F	0	A
10	F/M	0	F	1	F
11	F♯/halfdim	1	E	1	G
12	G/sus	1	G	0	G
13	G/sus	0	G	0	G
14	E♭/7	0	G	0	G
15	E♭/7	0	G	0	G
16	F♯/halfdim	1	E	0	G
17	F/m	1	C	1	F
18	C/M	0	C	0	F
19	B/7	1	G	1	B
20	G/M	0	G	1	D
21	A/7	1	F	0	D
22	F/M	0	F	1	F
23	A♭/sus	1	E♭	1	A♭
24	A♭/7	0	E♭	0	A♭

Continued on next page

Day Waves by Chick Corea

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
25	B♭/m	0	E♭	0	A♭
26	B♭/m	0	E♭	0	A♭
27	E♭/dim	0	E♭	1	C♯
28	E♭/M	0	E♭	1	B♭
29	E♭/M	0	E♭	0	B♭

Elizete by Claire Fischer

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	A/m	1	A	1	A
2	A/m7	0	A	0	A
3	D/m7	1	A	0	A
4	E/alt	0	A	0	A
5	E♭/M	1	C	1	E♭
6	D/m7	0	C	1	C
7	G/7	0	C	0	C
8	C/M	0	C	0	C
9	C/7	0	C	0	C
10	B/halfdim	1	A	1	A
11	E/7	0	A	0	A
12	E/halfdim	1	D	1	D
13	A/7	0	D	0	D
14	A/halfdim	1	G	1	G
15	D/7	0	G	0	G
16	B/halfdim	1	E	1	A
17	E/7	0	E	0	A
18	A/m	0	E	0	A
19	A/m7	0	E	0	A
20	D/m7	1	A	0	A
21	E/alt	0	A	0	A
22	E♭/M	1	C	1	E♭
23	D/m7	0	C	1	C
24	G/7	0	C	0	C
25	C/M	0	C	0	C
26	C/7	0	C	0	C
27	B/halfdim	1	A	1	A
28	E/7	0	A	0	A
29	A/m7	1	D	0	A
30	D/7	0	D	0	A
31	D♭/M	1	C	1	E♭
32	G/7	0	C	1	C
33	C/M	0	C	0	C

Continued on next page

Elizete by Claire Fischer					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
34	B/halfdim	1	A	1	A
35	E/7	0	A	0	A
36	A/m	0	A	0	A

Freddie the Freeloader by Miles Davis					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	B♭/7	1	B♭	1	B♭
2	B♭/7	0	B♭	0	B♭
3	B♭/7	0	B♭	0	B♭
4	B♭/7	0	B♭	0	B♭
5	E♭/7	1	E♭	0	B♭
6	E♭/7	0	E♭	0	B♭
7	B♭/7	1	B♭	0	B♭
8	B♭/7	0	B♭	0	B♭
9	F/7	1	F	0	B♭
10	B♭/7	1	E♭	0	B♭
11	E♭/7	0	E♭	0	B♭
12	A♭/7	0	E♭	1	A♭
13	A♭/7	0	E♭	0	A♭
14	B♭/7	1	B♭	1	B♭
15	B♭/7	0	B♭	0	B♭
16	B♭/7	0	B♭	0	B♭
17	B♭/7	0	B♭	0	B♭
18	E♭/7	1	E♭	0	B♭
19	E♭/7	0	E♭	0	B♭
20	B♭/7	1	B♭	0	B♭
21	B♭/7	0	B♭	0	B♭
22	F/7	1	F	0	B♭
23	B♭/7	1	E♭	0	B♭
24	E♭/7	0	E♭	0	B♭
25	B♭/7	1	B♭	0	B♭
26	B♭/7	0	B♭	0	B♭

Gary's Waltz by Gary McFarland					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	A/m7	1	A	1	G
2	A/m7	0	A	0	G
3	A/m7	0	A	0	G
4	A/m7	0	A	0	G
5	A/m7	1	A	0	G
6	A/m7	0	A	0	G
7	A/m7	0	A	0	G
8	A/m7	0	A	0	G
9	G/m7	1	G	0	G
10	G/m7	0	G	0	G
11	G/m7	0	G	0	G
12	G/m7	0	G	0	G
13	A/m7	1	A	1	A
14	B/m7	0	A	0	A
15	A/m7	0	A	0	A
16	A/m7	0	A	0	A
17	A/m7	0	A	0	A
18	A/m7	0	A	0	A
19	A/m7	1	A	0	A
20	A/m7	0	A	0	A
21	A/m7	0	A	0	A
22	A/m7	0	A	0	A
23	A/m7	1	A	0	A
24	A/m7	0	A	0	A
25	A/m7	0	A	0	A
26	A/m7	0	A	0	A
27	G/m7	1	G	1	G
28	G/m7	0	G	0	G
29	G/m7	0	G	0	G
30	G/m7	0	G	0	G
31	A/m7	1	A	1	A
32	B/m7	0	A	0	A
33	A/m7	0	A	0	A
34	A/m7	0	A	0	A
35	A/m7	0	A	0	A
36	A/m7	0	A	0	A
37	A/m7	1	A	0	A
38	A/m7	0	A	0	A
39	A♭/7	1	G	1	A♭
40	A♭/7	0	G	0	A♭
41	G/7	0	G	1	D
42	G/7	0	G	0	D

Continued on next page

Gary's Waltz by Gary McFarland

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
43	F♯/m7	1	F	0	D
44	F♯/m7	0	F	0	D
45	F/M	0	F	1	F
46	F/M	0	F	0	F
47	E/M	1	E	1	A♭
48	E/M	0	E	0	A♭
49	E♭/alt	1	D	0	A♭
50	E♭/alt	0	D	0	A♭
51	D/7	0	D	1	F♯
52	D/7	0	D	0	F♯
53	D♭/7	1	C♯	0	F♯
54	D♭/7	0	C♯	0	F♯
55	C/M	1	C	1	C
56	C/M	0	C	0	C
57	C/M	0	C	0	C
58	C/M	0	C	0	C
59	C/M	1	C	0	C
60	C/M	0	C	0	C
61	C/M	0	C	0	C
62	C/M	0	C	0	C

Gemini by Jimmy Heath

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	E♭/M	1	E♭	1	A♭
2	E♭/no3rd	0	E♭	0	A♭
3	E♭/M	0	E♭	0	A♭
4	E♭/no3rd	0	E♭	0	A♭
5	E♭/M	1	E♭	0	A♭
6	E♭/no3rd	0	E♭	0	A♭
7	E♭/M	0	E♭	0	A♭
8	E♭/no3rd	0	E♭	0	A♭
9	E♭/m7	1	A♭	1	E♭
10	A♭/7	0	A♭	0	E♭
11	E♭/m7	0	A♭	0	E♭
12	A♭/7	0	A♭	0	E♭
13	E♭/M	1	E♭	1	A♭
14	E♭/no3rd	0	E♭	0	A♭
15	E♭/M	0	E♭	0	A♭
16	G/alt	1	F	0	A♭
17	C/7	0	F	1	F
18	F/7	0	F	0	F

Continued on next page

Gemini by Jimmy Heath

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
19	F/7	0	F	0	F
20	F/7	0	F	0	F
21	B♭/alt	1	E♭	1	G
22	C/7	0	E♭	0	G
23	B♭/7	0	E♭	1	B♭
24	E♭/M	0	E♭	0	B♭
25	E♭/no3rd	0	E♭	1	A♭
26	E♭/M	0	E♭	0	A♭
27	E♭/no3rd	0	E♭	0	A♭

Giant Steps by John Coltrane

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	B/M	1	B	1	B
2	D/7	1	G	1	G
3	G/M	0	G	0	G
4	B♭/7	1	E♭	1	E♭
5	E♭/M	0	E♭	0	E♭
6	A/m7	1	G	1	G
7	D/7	0	G	0	G
8	G/M	0	G	0	G
9	B♭/7	1	E♭	1	E♭
10	E♭/M	0	E♭	0	E♭
11	F♯/7	1	B	1	B
12	B/M	0	B	0	B
13	F/m7	1	E♭	1	E♭
14	B♭/7	0	E♭	0	E♭
15	E♭/M	0	E♭	0	E♭
16	A/m7	1	G	1	G
17	D/7	0	G	0	G
18	G/M	0	G	0	G
19	D♭/m7	1	B	1	B
20	F♯/7	0	B	0	B
21	B/M	0	B	0	B
22	F/m7	1	E♭	1	E♭
23	B♭/7	0	E♭	0	E♭
24	E♭/M	0	E♭	0	E♭
25	D♭/m7	1	B	1	B
26	F♯/7	0	B	0	B
27	B/M	0	B	0	B

Half Nelson by Miles David

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	C/M	1	C	1	F
2	C/M	0	C	0	F
3	F/m7	1	E♭	0	F
4	B♭/7	0	E♭	0	F
5	F/m7	1	E♭	0	F
6	B♭/7	0	E♭	0	F
7	C/M	1	C	1	C
8	D/m7	0	C	0	C
9	G/7	0	C	0	C
10	C/M	0	C	0	C
11	B/m7	1	A	1	B
12	E/7	0	A	0	B
13	B♭/m7	1	A♭	1	A♭
14	E♭/7	0	A♭	0	A♭
15	A♭/M	0	A♭	0	A♭
16	A♭/M	0	A♭	0	A♭
17	A/m7	1	G	1	A
18	D/7	0	G	0	A
19	A/m7	1	G	0	A
20	D/7	0	G	0	A
21	D/m7	1	C	1	C
22	G/7	0	C	0	C
23	C/M	0	C	0	C
24	E♭/M	1	A♭	1	A♭
25	A♭/M	0	A♭	0	A♭
26	D♭/M	0	A♭	0	A♭

Jelly Roll by Charles Mingus

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	E♭/7	1	A♭	1	A♭
2	E♭/7	0	A♭	0	A♭
3	E♭/7	0	A♭	0	A♭
4	E♭/7	0	A♭	0	A♭
5	A♭/7	1	C♯	0	A♭
6	A♭/7	0	C♯	0	A♭
7	D♭/7	1	F♯	0	A♭
8	D♭/7	0	F♯	0	A♭
9	A♭/7	1	B♭	0	A♭
10	G/alt	0	B♭	1	E
11	F♯/7	0	B♭	1	B♭
12	F/7	0	B♭	0	B♭

Continued on next page

Jelly Roll by Charles Mingus

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
13	B♭/m7	1	A♭	0	B♭
14	E♭/7	0	A♭	1	A♭
15	A♭/7	0	A♭	0	A♭
16	A♭/7	0	A♭	0	A♭
17	A♭/7	0	A♭	0	A♭
18	A/7	1	A♭	1	A
19	A♭/7	0	A♭	1	A♭

Jinrikisha by Joe Henderson

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	D♭/M	1	C	1	A♭
2	D♭/M	0	C	0	A♭
3	C/sus	0	C	0	A♭
4	C/sus	0	C	0	A♭
5	B♭/m7	1	A♭	0	A♭
6	B♭/m7	0	A♭	0	A♭
7	A♭/M	0	A♭	0	A♭
8	A♭/M	0	A♭	0	A♭
9	F♯/M	1	F	1	F♯
10	F♯/M	0	F	0	F♯
11	F/m	0	F	1	F
12	F/m	0	F	0	F
13	F/m	0	F	0	F
14	F/m	0	F	0	F
15	G/halfdim	1	F	0	F
16	C/7	0	F	0	F
17	D♭/M	1	C	1	A♭
18	D♭/M	0	C	0	A♭
19	C/sus	0	C	0	A♭
20	C/sus	0	C	0	A♭
21	B♭/m7	1	A♭	0	A♭
22	B♭/m7	0	A♭	0	A♭
23	A♭/M	0	A♭	0	A♭
24	A♭/M	0	A♭	0	A♭
25	F♯/M	1	F	1	C♯
26	F♯/M	0	F	0	C♯
27	F/m	0	F	0	C♯
28	F/m	0	F	0	C♯
29	F/m	0	F	1	A♭
30	F/m	0	F	0	A♭
31	B♭/m7	1	F♯	0	A♭

Continued on next page

Jinrikisha by Joe Henderson					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
32	F♯/M	0	F♯	0	A♭
33	B/M	0	F♯	0	A♭
34	F♯/M	0	F♯	0	A♭
35	G/halfdim	1	F	1	C♯
36	C/alt	0	F	0	C♯

Lonnie's Lament by John Coltrane					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	C/m7	1	C	1	B♭
2	D/m7	0	C	0	B♭
3	E♭/M	0	C	0	B♭
4	D/m7	0	C	0	B♭
5	C/m7	1	C	0	B♭
6	D/m7	0	C	0	B♭
7	E♭/M	0	C	0	B♭
8	D/m7	0	C	0	B♭
9	C/m7	1	C	0	B♭
10	D/m7	0	C	0	B♭
11	E♭/M	0	C	0	B♭
12	D/m7	0	C	0	B♭
13	C/m7	1	C	0	B♭
14	D/m7	0	C	0	B♭
15	E♭/M	0	C	0	B♭
16	D/m7	0	C	0	B♭
17	C/m7	0	C	0	B♭
18	B♭/7	1	E♭	0	B♭
19	E♭/M	0	E♭	0	B♭
20	A♭/M	0	E♭	1	C♯
21	A♭/7	1	C♯	0	C♯
22	A/7	0	C♯	0	C♯
23	A♭/7	1	C	0	C♯
24	G/alt	0	C	1	C
25	C/m7	0	C	0	C
26	D/m7	0	C	0	C
27	E♭/M	0	C	1	E♭
28	G/m7	1	C	0	C
29	G/7	0	C	1	C
30	C/m7	0	C	1	B♭
31	D/m7	0	C	0	B♭
32	E♭/M	0	C	0	B♭
33	D/m7	0	C	0	B♭

Continued on next page

Lonnie's Lament by John Coltrane

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
34	C/m7	0	C	0	B♭
35	C/m7	0	C	0	B♭

Lullaby of Birdland by George Gershwin

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
1	F/m	1	F	1	F
2	G/7	1	C	1	G
3	C/7	1	F	0	G
4	F/m	0	F	1	A♭
5	B♭/m7	1	A♭	0	A♭
6	E♭/7	0	A♭	0	A♭
7	A♭/M	0	A♭	0	A♭
8	F/m7	1	A♭	0	A♭
9	B♭/m7	0	A♭	0	A♭
10	E♭/7	0	A♭	0	A♭
11	A♭/M	0	A♭	0	A♭
12	D♭/7	1	F	0	A♭
13	C/7	0	F	1	F
14	F/m	0	F	1	C
15	G/7	1	C	1	G
16	C/7	1	F	0	G
17	F/m	0	F	1	C♯
18	B♭/m7	1	A♭	0	A♭
19	E♭/7	0	A♭	0	A♭
20	A♭/M	0	A♭	0	A♭
21	F/m7	1	A♭	0	A♭
22	B♭/m7	0	A♭	0	A♭
23	E♭/7	0	A♭	0	A♭
24	A♭/M	0	A♭	0	A♭
25	E♭/7	1	A♭	0	A♭
26	A♭/M	0	A♭	1	G
27	F/7	1	B♭	1	A♭
28	B♭/m7	0	B♭	0	A♭
29	B♭/m7	1	A♭	0	A♭
30	E♭/7	0	A♭	0	A♭
31	A♭/M	0	A♭	0	A♭
32	F/7	1	B♭	0	A♭
33	B♭/m7	0	B♭	1	G
34	B♭/m7	1	A♭	1	A♭
35	E♭/7	0	A♭	0	A♭
36	A♭/M	0	A♭	0	A♭

Continued on next page

Lullaby of Birdland by George Gershwin

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
37	C/7	1	F	0	A♭
38	F/m	0	F	0	A♭
39	G/7	1	C	0	A♭
40	C/7	1	F	0	A♭
41	F/m	0	F	0	A♭
42	B♭/m7	1	A♭	0	A♭
43	E♭/7	0	A♭	0	A♭
44	A♭/M	0	A♭	1	F
45	F/m7	1	A♭	0	F
46	B♭/m7	0	A♭	1	A♭
47	E♭/7	0	A♭	0	A♭
48	A♭/M	0	A♭	0	A♭
49	E♭/7	1	A♭	0	A♭
50	A♭/M	0	A♭	1	F

Mahjong by Wayne Shorter

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
	chords	hand.boundaries	hand.labels	pachet.boundaries	pachet.labels
1	F/m7	1	F	1	F
2	F/m7	0	F	0	F
3	F/m7	0	F	0	F
4	F/m7	0	F	0	F
5	F/m7	0	F	0	F
6	F/m7	0	F	0	F
7	F/m7	0	F	0	F
8	F/m7	0	F	0	F
9	F/m7	0	F	0	F
10	F/m7	0	F	0	F
11	F/m7	0	F	0	F
12	F/m7	0	F	0	F
13	F/m7	0	F	0	F
14	F/m7	0	F	0	F
15	F/m7	0	F	0	F
16	F/m7	0	F	0	F
17	D♭/M	1	C♯	1	C♯
18	D♭/M	0	C♯	0	C♯
19	D♭/M	0	C♯	0	C♯
20	D♭/M	0	C♯	0	C♯
21	D♭/M	0	C♯	0	C♯
22	D♭/M	0	C♯	0	C♯
23	D♭/M	0	C♯	0	C♯

Continued on next page

Mahjong by Wayne Shorter

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
24	D♭/M	0	C♯	0	C♯
25	D/7	1	D	1	D
26	E♭/m7	1	C♯	1	C♯
27	A♭/7	0	C♯	0	C♯
28	D♭/M	0	C♯	0	C♯
29	D♭/m7	1	B	1	C♯
30	F♯/7	0	B	0	C♯
31	F/m7	1	F	1	E♭
32	F/m7	0	F	0	E♭
33	F/m7	0	F	0	E♭
34	F/m7	0	F	0	E♭
35	F/m7	0	F	0	E♭
36	F/m7	0	F	0	E♭
37	F/m7	0	F	0	E♭
38	F/m7	0	F	0	E♭

My Romance by Rodgers/Hart

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	B♭/M	1	B♭	1	B♭
2	C/m7	0	B♭	0	B♭
3	D/m7	1	C	0	B♭
4	D♭/dim	0	C	1	B
5	C/m7	1	B♭	1	B♭
6	F/7	0	B♭	0	B♭
7	B♭/M	0	B♭	0	B♭
8	D/7	1	G	1	G
9	G/m	0	G	0	G
10	G/m	0	G	0	G
11	G/m7	0	G	0	G
12	G/7	1	C	1	C
13	C/m7	0	C	0	C
14	F/7	1	B♭	0	C
15	B♭/M	0	B♭	1	E♭
16	B♭/7	1	E♭	0	E♭
17	E♭/M	0	E♭	0	E♭
18	A♭/7	1	B♭	0	E♭
19	B♭/M	0	B♭	0	E♭
20	B♭/7	1	E♭	0	E♭
21	E♭/M	0	E♭	0	E♭
22	A♭/7	1	B♭	0	E♭
23	B♭/M	0	B♭	0	E♭

Continued on next page

My Romance by Rodgers/Hart					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
24	E/halfdim	1	D	1	D
25	A/7	0	D	0	D
26	D/m7	0	D	0	D
27	D♭/7	1	C	1	F
28	C/sus	0	C	0	F
29	C/7	1	G	0	F
30	C/m7	1	B♭	1	B♭
31	F/7	0	B♭	0	B♭
32	B♭/M	0	B♭	0	B♭
33	C/m7	0	B♭	0	B♭
34	D/m7	1	C	0	B♭
35	D♭/dim	0	C	1	B
36	C/m7	1	B♭	1	B♭
37	F/7	0	B♭	0	B♭
38	B♭/M	0	B♭	0	B♭
39	D/7	1	G	1	G
40	G/m	0	G	0	G
41	G/m	0	G	0	G
42	G/m7	0	G	0	G
43	G/7	1	C	1	G
44	C/m7	0	C	1	B♭
45	F/7	1	B♭	0	B♭
46	F/m7	1	E♭	0	B♭
47	B♭/7	0	E♭	0	B♭
48	E♭/M	0	E♭	0	B♭
49	G/7	1	C	1	B
50	C/m7	0	C	1	B♭
51	C/m7	1	G	0	B♭
52	A/halfdim	0	G	0	B♭
53	D/7	0	G	1	G
54	G/m7	0	G	0	G
55	F♯/7	1	B♭	0	G
56	B♭/M	0	B♭	0	G
57	C/m7	1	B♭	1	G
58	F/7	0	B♭	1	B♭
59	B♭/M	0	B♭	0	B♭
60	C/m7	1	B♭	0	B♭
61	F/7	0	B♭	0	B♭
62	B♭/M	0	B♭	0	B♭

Pent-Up House by Sonny Rollins

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	A/m7	1	G	1	C♯
2	A♭/7	0	G	0	C♯
3	A/m7	1	G	0	C♯
4	A♭/7	0	G	0	C♯
5	G/M	0	G	1	C
6	A♭/7	1	G	0	C
7	G/M	0	G	0	C
8	A/m7	1	G	1	C♯
9	A♭/7	0	G	0	C♯
10	A/m7	1	G	0	C♯
11	A♭/7	0	G	0	C♯
12	G/M	0	G	1	C
13	A♭/7	1	G	0	C
14	G/M	0	G	0	C
15	D/m7	1	C	1	F♯
16	D♭/7	0	C	0	F♯
17	D/m7	1	C	0	F♯
18	D♭/7	0	C	0	F♯
19	C/m7	0	C	1	C
20	C/m7	1	B♭	0	C
21	F/7	0	B♭	0	C
22	A/m7	1	G	1	C♯
23	A♭/7	0	G	0	C♯
24	A/m7	1	G	0	C♯
25	A♭/7	0	G	0	C♯
26	G/M	0	G	1	C
27	A♭/7	1	G	0	C
28	G/M	0	G	1	G

Resolution by Mahavishnu

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	A/dim	1	B♭	1	E♭
2	A/dim	0	B♭	0	E♭
3	B♭/M	0	B♭	0	E♭
4	B♭/M	0	B♭	0	E♭
5	A/m	1	D	1	A
6	A/m	0	D	0	A
7	D/M	0	D	0	A
8	D/M	0	D	0	A
9	A/m	1	B♭	0	A
10	A/m	0	B♭	0	A

Continued on next page

Resolution by Mahavishnu

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
11	B♭/M	0	B♭	1	E♭
12	B♭/M	0	B♭	0	E♭
13	A/dim	1	B♭	0	E♭
14	A/dim	0	B♭	0	E♭
15	B♭/M	0	B♭	0	E♭
16	B♭/M	0	B♭	0	E♭
17	A/m	1	D	1	A
18	A/m	0	D	0	A
19	D/M	0	D	0	A
20	D/M	0	D	0	A
21	A/m	1	B♭	0	A
22	A/m	0	B♭	0	A
23	B♭/M	0	B♭	1	E♭
24	B♭/M	0	B♭	0	E♭
25	A/dim	1	B♭	0	E♭
26	A/dim	0	B♭	0	E♭
27	B♭/M	0	B♭	0	E♭
28	B♭/M	0	B♭	0	E♭
29	A/m	1	D	1	A
30	A/m	0	D	0	A
31	D/M	0	D	0	A
32	D/M	0	D	0	A
33	A/m	1	B♭	0	A
34	A/m	0	B♭	0	A
35	B♭/M	0	B♭	1	E♭
36	B♭/M	0	B♭	0	E♭
37	A/dim	1	A	0	E♭

Saga of Harrison Crabfeathers by Steve Kuhn

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	E/m	1	E	1	E
2	E/m	0	E	0	E
3	E/m	0	E	0	E
4	E/m	0	E	0	E
5	C/M	0	E	0	E
6	C/M	0	E	0	E
7	C/M	0	E	0	E
8	C/M	0	E	0	E
9	A/m	1	E	0	E
10	A/m	0	E	0	E
11	A/m	0	E	0	E

Continued on next page

Saga of Harrison Crabfeathers by Steve Kuhn

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
12	A/m	0	E	0	E
13	E/m	0	E	0	E
14	E/m	0	E	0	E
15	E/m	0	E	0	E
16	E/m	0	E	0	E
17	D/m	1	D	1	D
18	D/m	0	D	0	D
19	D/m	0	D	0	D
20	D/m	0	D	0	D
21	B♭/M	0	D	0	D
22	B♭/M	0	D	0	D
23	B♭/M	0	D	0	D
24	B♭/M	0	D	0	D
25	G/m	1	D	0	D
26	G/m	0	D	0	D
27	G/m	0	D	0	D
28	G/m	0	D	0	D
29	D/m	0	D	0	D
30	D/m	0	D	0	D
31	D/m	0	D	0	D
32	D/m	0	D	0	D
33	A♭/M	1	E♭	1	A♭
34	A♭/M	0	E♭	0	A♭
35	A♭/M	0	E♭	0	A♭
36	A♭/M	0	E♭	0	A♭
37	A♭/M	0	E♭	0	A♭
38	A♭/M	0	E♭	0	A♭
39	A♭/M	0	E♭	0	A♭
40	A♭/M	0	E♭	0	A♭
41	C/m	1	C	0	A♭
42	C/m	0	C	0	A♭
43	C/m	0	C	0	A♭
44	C/m	0	C	0	A♭
45	A♭/M	0	C	0	A♭
46	A♭/M	0	C	0	A♭
47	A♭/M	0	C	0	A♭
48	A♭/M	0	C	0	A♭
49	F/m	1	C	0	A♭
50	F/m	0	C	0	A♭
51	F/m	0	C	0	A♭
52	F/m	0	C	0	A♭
53	C/m	0	C	0	A♭

Continued on next page

Saga of Harrison Crabfeathers by Steve Kuhn					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
54	C/m	0	C	0	A♭
55	C/m	0	C	0	A♭
56	C/m	0	C	0	A♭
57	E/m7	1	E	1	E
58	E/m7	0	E	0	E
59	E/m7	0	E	0	E
60	E/m7	0	E	0	E
61	E/m7	0	E	0	E
62	E/m7	0	E	0	E
63	E/m7	0	E	0	E
64	E/m7	0	E	0	E
65	C/M	1	C	0	E
66	C/M	0	C	0	E
67	C/M	0	C	0	E
68	C/M	0	C	0	E
69	E/m7	1	E	0	E
70	E/m7	0	E	0	E
71	E/m7	0	E	0	E
72	E/m7	0	E	0	E
73	D/m7	1	D	1	D
74	D/m7	0	D	0	D
75	D/m7	0	D	0	D
76	D/m7	0	D	0	D
77	D/m7	0	D	0	D
78	D/m7	0	D	0	D
79	D/m7	0	D	0	D
80	D/m7	0	D	0	D
81	B♭/M	1	B♭	0	D
82	B♭/M	0	B♭	0	D
83	B♭/M	0	B♭	0	D
84	B♭/M	0	B♭	0	D
85	D/m7	1	B♭	0	D
86	D/m7	0	B♭	0	D
87	D/m7	0	B♭	0	D
88	D/m7	0	B♭	0	D
89	A♭/M	1	A♭	1	A♭
90	A♭/M	0	A♭	0	A♭
91	A♭/M	0	A♭	0	A♭
92	A♭/M	0	A♭	0	A♭
93	A♭/M	0	A♭	0	A♭
94	A♭/M	0	A♭	0	A♭
95	A♭/M	0	A♭	0	A♭

Continued on next page

Saga of Harrison Crabfeathers by Steve Kuhn

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
96	A♭/M	0	A♭	0	A♭
97	C/m7	1	C	0	A♭
98	C/m7	0	C	0	A♭
99	C/m7	0	C	0	A♭
100	C/m7	0	C	0	A♭
101	C/m7	0	C	0	A♭
102	C/m7	0	C	0	A♭
103	C/m7	0	C	0	A♭
104	C/m7	0	C	0	A♭
105	A♭/M	1	A♭	0	A♭
106	A♭/M	0	A♭	0	A♭
107	A♭/M	0	A♭	0	A♭
108	A♭/M	0	A♭	0	A♭
109	C/m7	1	A♭	0	A♭
110	C/m7	0	A♭	0	A♭
111	C/m7	0	A♭	0	A♭
112	C/m7	0	A♭	0	A♭

Scotch 'n' Soda by Dave Guard

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	A♭/M	1	A♭	1	A♭
2	D♭/7	0	A♭	0	A♭
3	E♭/6	1	E	0	A♭
4	G/m7	1	F	1	F
5	C/7	0	F	0	F
6	F/7	0	F	0	F
7	F/m7	1	E	1	F
8	B♭/7	0	E	1	B♭
9	D/m	1	C	0	B♭
10	A♭/m	0	C	1	C
11	G/7	0	C	1	A♭
12	A♭/M	1	A♭	0	A♭
13	D♭/7	0	A♭	0	A♭
14	E♭/6	1	E	1	F
15	G/m7	1	F	0	F
16	C/7	0	F	0	F
17	F/7	0	F	0	F
18	F/m7	1	E	1	F
19	B♭/7	0	E	0	F
20	E♭/7	1	A♭	1	A♭
21	B♭/m7	0	A♭	0	A♭

Continued on next page

Scotch ‘n’ Soda by Dave Guard

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
22	E♭/7	0	A♭	0	A♭
23	E♭/alt	0	A♭	0	A♭
24	A♭/M	0	A♭	0	A♭
25	A♭/M	0	A♭	1	F
26	E♭/M	0	A♭	0	F
27	F/m	1	E	0	F
28	B♭/7	0	E	0	F
29	E♭/M	0	E	1	F
30	F/7	1	B♭	1	A♭
31	F/7	0	B♭	0	A♭
32	B♭/7	0	B♭	0	A♭
33	F/m7	1	E♭	1	F
34	B♭/7	0	E♭	0	F
35	A♭/M	1	A♭	0	F
36	D♭/7	0	A♭	1	E♭
37	E♭/6	1	E	0	E♭
38	G/m7	1	F	1	A♭
39	C/7	0	F	0	A♭
40	F/7	0	F	1	E♭
41	F/m7	1	E♭	0	E♭
42	B♭/7	0	E♭	0	E♭
43	G/m7	1	F	0	E♭
44	C/7	0	F	0	E♭
45	F/m7	1	E♭	1	F
46	B♭/7	0	E♭	0	F
47	A♭/7	0	E♭	0	F
48	E♭/M	0	E♭	0	F

Shades of Light by Hubert Laws

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
1	A/m7	1	G	1	A
2	D/7	1	G	0	A
3	F/m7	1	E♭	1	F
4	B♭/7	0	E♭	0	F
5	F♯/7	1	C♯	1	F♯
6	G/7	0	C♯	0	F♯
7	A♭/7	0	C♯	1	C
8	A/m7	1	G	1	G
9	D/7	0	G	0	G
10	F/m7	1	E♭	1	F
11	B♭/7	0	E♭	0	F

Continued on next page

Shades of Light by Hubert Laws

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
12	B/m7	1	A	1	B
13	E/7	0	A	0	B
14	E♭/7	1	C♯	1	E♭
15	A♭/M	0	C♯	0	E♭
16	G/7	1	E	1	G
17	F♯/m7	0	E	1	E
18	B/7	0	E	0	E
19	E/M	0	E	0	E
20	F♯/m7	1	F♯	0	E
21	A♭/m7	0	F♯	0	E
22	A/m7	1	C	1	C
23	D/m7	0	C	0	C
24	G/7	0	C	0	C
25	C/M	0	C	0	C
26	E/7	1	A	1	E
27	A/m7	0	A	1	A

Skating in Central Park by John Lewis

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
1	C/M	1	C	1	C
2	G/7	0	C	0	C
3	C/M	0	C	0	C
4	G/7	0	C	0	C
5	D/m7	1	C	0	C
6	G/7	0	C	0	C
7	C/M	0	C	0	C
8	C/alt	1	F	1	A
9	F/M	0	F	0	A
10	B/7	1	E	1	E
11	E/m7	0	E	0	E
12	A/m7	0	E	0	E
13	D/m7	1	C	1	C
14	G/7	0	C	0	C
15	C/M	0	C	0	C
16	G/7	1	C	0	C
17	C/M	0	C	0	C
18	G/7	0	C	0	C
19	C/M	0	C	0	C
20	G/7	0	C	0	C
21	D/m7	1	C	0	C
22	G/7	0	C	0	C

Continued on next page

Skating in Central Park by John Lewis					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
23	C/M	0	C	0	C
24	C/alt	1	F	1	A
25	F/M	0	F	0	A
26	B/7	1	E	1	B
27	E/m7	0	E	1	C
28	A/m7	0	E	0	C
29	D/m7	1	C	0	C
30	G/7	0	C	0	C
31	C/M	0	C	0	C
32	C/alt	1	F	0	C
33	F/m	0	F	0	C
34	F/m	0	F	0	C
35	F/m7	1	E	0	C
36	D/halfdim	0	E	0	C
37	E♭/M	0	E	0	C
38	E♭/M	0	E	0	C
39	C/m7	1	G	1	A
40	E♭/M	0	G	0	A
41	A/m	0	G	1	E
42	A/m7	1	G	0	E
43	F♯/halfdim	0	G	0	E
44	F♯/halfdim	0	G	1	C
45	F/m7	1	E♭	0	C
46	F/m7	0	E♭	0	C
47	D/m7	1	C	0	C
48	G/7	0	C	0	C
49	C/M	0	C	0	C

Someday my Prince will Come by Frank Churchill					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	B♭/M	1	B♭	1	B♭
2	D/alt	1	G	1	G
3	E♭/M	0	G	0	G
4	G/alt	0	G	1	C
5	C/m7	1	C	0	C
6	G/alt	0	C	0	C
7	C/7	1	F	1	C
8	F/7	0	F	0	C
9	D/m7	1	D	0	C
10	D♭/dim	0	D	1	B
11	C/m7	1	B♭	1	C

Continued on next page

Someday my Prince will Come by Frank Churchill					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
12	F/7	0	B♭	0	C
13	D/m7	1	D	0	C
14	D♭/dim	0	D	1	B
15	C/m7	1	B♭	1	B♭
16	F/7	0	B♭	0	B♭
17	B♭/M	0	B♭	1	B♭
18	D/alt	1	G	1	G
19	E♭/M	0	G	0	G
20	G/alt	0	G	1	G
21	C/m7	1	C	0	G
22	G/alt	0	C	0	G
23	C/7	1	F	1	F
24	F/7	0	F	0	F
25	F/m7	1	E♭	0	F
26	B♭/7	0	E♭	1	G
27	E♭/M	0	E♭	0	G
28	E/dim	1	B♭	1	C
29	B♭/M	0	B♭	0	C
30	F/sus	1	B♭	0	C
31	F/7	0	B♭	0	C
32	B♭/M	0	B♭	1	F
33	B♭/M	0	B♭	0	F

The Song is You by Kern/Hammerstein					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	C/M	1	C	1	E
2	C/dim	0	C	0	E
3	D/m7	0	C	1	D
4	G/7	0	C	0	D
5	E/m7	1	D	0	D
6	A/7	0	D	0	D
7	D/m7	1	C	0	D
8	G/7	0	C	0	D
9	C/M	0	C	1	C
10	E/m	1	C	0	C
11	D/m7	0	C	0	C
12	G/7	0	C	1	C
13	D/halfdim	1	C	0	C
14	G/7	0	C	1	D
15	E/m7	1	D	0	D
16	A/7	0	D	0	D

Continued on next page

The Song is You by Kern/Hammerstein

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
17	D/m7	1	C	0	D
18	G/7	0	C	0	D
19	C/M	0	C	1	E
20	C/dim	1	C	0	E
21	D/m7	0	C	1	D
22	G/7	0	C	0	D
23	E/m7	1	D	0	D
24	A/7	0	D	0	D
25	D/m7	1	C	0	D
26	G/7	0	C	0	D
27	C/M	1	C	1	E
28	A/7	1	D	0	E
29	D/m7	0	D	1	D
30	G/7	1	C	0	D
31	C/6	0	C	0	D
32	C/6	0	C	0	D
33	E/M	1	E	1	E
34	F♯/m7	0	E	1	D
35	B/7	0	E	0	D
36	E/M	0	E	0	D
37	B♭/m7	1	A♭	0	D
38	E♭/7	0	A♭	0	D
39	A♭/m7	1	F♯	1	G
40	D♭/7	0	F♯	1	C
41	F♯/7	1	B	0	C
42	B/7	1	E	0	C
43	G/7	0	C	0	C
44	C/M	1	C	1	E
45	C/dim	1	C	0	E
46	D/m7	0	C	0	E
47	G/7	0	C	0	E
48	C/M	0	C	0	E
49	C/7	1	F	0	E
50	F/M	0	F	1	A♭
51	F/m	1	D	0	A♭
52	E/m7	0	D	1	F♯
53	A/7	0	D	0	F♯
54	D/m7	1	C	0	F♯
55	G/7	0	C	0	F♯
56	C/6	0	C	0	F♯
57	D/m7	1	C	0	G
58	G/7	0	C	0	G

Continued on next page

The Song is You by Kern/Hammerstein					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
59	C/M	0	C	0	C

Three Flowers by McCoy Tyner					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	E♭/M	1	E♭	1	A♭
2	D♭/M	0	E♭	0	A♭
3	E♭/M	0	E♭	0	A♭
4	D♭/M	0	E♭	0	A♭
5	E♭/M	0	E♭	0	A♭
6	D♭/M	0	E♭	0	A♭
7	A/m7	1	G	1	G
8	D/7	0	G	0	G
9	G/M	1	G	0	G
10	F/7	0	G	1	G
11	G/M	0	G	0	G
12	F/7	0	G	0	G
13	E/M	1	E	1	A
14	D/7	0	E	0	A
15	E/M	0	E	0	A
16	F/m7	1	E♭	1	E♭
17	B♭/7	0	E♭	0	E♭
18	E♭/M	1	E♭	0	E♭
19	D♭/M	0	E♭	1	A♭
20	E♭/M	0	E♭	0	A♭
21	D♭/M	0	E♭	0	A♭
22	E♭/M	0	E♭	0	A♭
23	D♭/M	0	E♭	0	A♭
24	A/m7	1	G	1	A
25	D/7	0	G	0	A
26	G/M	1	G	1	A♭
27	F/7	0	G	0	A♭
28	G/M	0	G	0	A♭
29	F/7	0	G	0	A♭
30	E/M	1	E	0	A♭
31	D/7	0	E	0	A♭
32	E/M	0	E	1	A
33	F/m7	1	E♭	0	A
34	B♭/7	0	E♭	0	A♭
35	E♭/M	0	E♭	0	A♭

What am I Here For by Duke Ellington

Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
1	C/M	1	C	1	F
2	D♭/dim	1	D	0	F
3	D/m7	1	C	1	C
4	G/7	0	C	0	C
5	C/M	0	C	0	C
6	D♭/dim	1	D	1	D
7	D/m7	1	C	0	D
8	G/7	0	C	0	D
9	G/m7	1	F	1	F
10	C/7	0	F	0	F
11	F/M	0	F	0	F
12	E/7	1	A	1	A
13	A/m7	0	A	0	A
14	A/m7	1	D	0	A
15	D/7	0	D	0	A
16	D/m7	1	C	1	D
17	D♭/7	0	C	1	F
18	C/M	0	C	0	F
19	D♭/dim	1	D	0	F
20	D/m7	1	C	1	C
21	G/7	0	C	0	C
22	C/M	0	C	0	C
23	D♭/dim	1	D	1	D
24	D/m7	1	C	0	D
25	G/7	0	C	0	D
26	G/m7	1	F	1	F
27	C/7	0	F	0	F
28	F/M	0	F	0	F
29	B♭/7	0	F	0	F
30	C/M	1	C	0	F
31	D♭/dim	1	D	1	D
32	D/m7	1	C	0	D
33	G/7	0	C	0	D
34	F♯/halfdim	1	E	1	E
35	B/7	0	E	0	E
36	B/7	0	E	0	E
37	F/m7	1	F	1	F
38	D/7	1	C	1	F♯
39	D/7	0	C	0	F♯
40	D♭/M	0	C	0	F♯
41	D♭/M	0	C	0	F♯
42	C/M	0	C	1	C

You Took Advantage of Me by Rodgers/Hart

Event	Chord	Boundary	Chunk label	Boundary	Chunk label
ID		(hand labelled)	(hand labelled)	(Pachet, 2000)	(Pachet, 2000)
1	E♭/M	1	E♭	1	E♭
2	E/dim	0	E♭	1	F
3	F/m7	1	E♭	0	F
4	B♭/7	0	E♭	0	F
5	G/m7	1	G	0	F
6	F♯/dim	0	G	1	E
7	F/m7	1	E♭	1	E♭
8	B♭/7	0	E♭	0	E♭
9	E♭/M	0	E♭	0	E♭
10	E♭/7	1	A♭	0	E♭
11	A♭/M	0	A♭	0	E♭
12	A♭/m	0	A♭	1	A♭
13	E♭/M	1	E♭	1	B♭
14	B♭/7	0	E♭	0	B♭
15	E♭/M	0	E♭	0	B♭
16	B♭/7	1	C	0	B♭
17	C/m	0	C	0	B♭
18	D/7	1	C	1	G
19	G/7	0	C	0	G
20	C/7	1	F	0	G
21	F/7	1	E♭	1	B♭
22	B♭/7	0	E♭	0	B♭
23	E♭/M	0	E♭	0	B♭
24	C/m	0	E♭	0	B♭
25	D/7	1	C	1	G
26	G/7	0	C	0	G
27	C/7	1	F	0	G
28	F/7	1	E♭	1	F
29	B♭/7	0	E♭	0	F
30	F/m7	1	E♭	1	E♭
31	B♭/7	0	E♭	0	E♭
32	E♭/M	0	E♭	0	E♭
33	E/dim	0	E♭	1	F
34	F/m7	1	E♭	0	F
35	B♭/7	0	E♭	0	F
36	G/m7	1	G	0	F
37	F♯/dim	0	G	1	E
38	F/m7	1	E♭	1	E♭
39	B♭/7	0	E♭	0	E♭
40	E♭/M	0	E♭	0	E♭
41	E♭/7	1	A♭	0	E♭
42	A♭/M	0	A♭	0	E♭

Continued on next page

You Took Advantage of Me by Rodgers/Hart					
Event ID	Chord	Boundary (hand labelled)	Chunk label (hand labelled)	Boundary (Pachet, 2000)	Chunk label (Pachet, 2000)
43	A♭/m	0	A♭	1	A♭
44	E♭/M	1	E♭	1	E♭
45	B♭/7	0	E♭	0	E♭
46	E♭/M	0	E♭	0	E♭

Author's Publications

Hedges, T. W. & McPherson, A. (2013). 3D gestural interaction with harmonic pitch space. In *Sound and music computing conference: SMC 2013* (pp. 103–108). Stockholm, Sweden.

Hedges, T. W. & Rohrmeier, M. A. (2011). Exploring Rameau and beyond: A corpus study of root progression theories. In C. Agon, E. Amiot, M. Andreatta, G. Assayag, J. Bresson & J. Manderau (Eds.), *Proceedings of mathematics and computation in music, third international conference: MCM 2011* (pp. 334–337). Berlin, Heidelberg: Springer.

Hedges, T. W., Roy, P. & Pachet, F. (2014). Predicting the composer and style of jazz chord progressions. *Journal of New Music Research*, *43*(3), 276–290.

Hedges, T. W. & Wiggins, G. A. (2015). *Segmentation and grouping structures in jazz chord sequences: An information-theoretic approach.* Paper presented at Speech Language, Music, Art, Reasoning Thought Conference: SMART 2015. Amsterdam, Netherlands.

Hedges, T. W. & Wiggins, G. A. (2016a). Improving predictions of derived viewpoints in multiple viewpoint systems. In *Proceedings of 17th international society for music information retrieval conference, ISMIR 2016* (pp. 420–426). New York, NY.

Hedges, T. W. & Wiggins, G. A. (2016b). The prediction of merged attributes with multiple viewpoint systems. *Journal of New Music Research*, 1–19.

Bibliography

Abdallah, S. M., Gold, N. & Marsden, A. (2015). Analysing symbolic music with probabilistic grammars. In D. Meredith (Ed.), *Computational music analysis* (pp. 157–189). Switzerland.

Abdallah, S. M. & Plumbley, M. D. (2009). Information dynamics: Patterns of expectation and surprise in the perception of music. *Connection Science*, *21*(2-3), 89–117.

Agres, K., Abdallah, S. M. & Pearce, M. (2017). Information-theoretic properties of auditory sequences dynamically influence expectation and memory. *Cognitive Science*, *21*(5), 89–34.

Allan, M. & Williams, C. K. I. (2005). Harmonising chorales by probabilistic inference. *Advances in Neural Information Processing Systems*, *17*, 25–32.

Assayag, G. & Dubnov, S. (2004). Using factor oracles for machine improvisation. *Soft Computing*, *8*(9), 1–7.

Attneave, F. & Olson, R. K. (1971). Pitch as a medium: A new approach to psychophysical scaling. *The American Journal of Psychology*, *84*(2), 147–166.

Ayotte, J., Peretz, I. & Hyde, K. (2002). Congenital amusia: A group study of adults afflicted with a music-specific disorder. *Brain: A Journal of Neurology*, *125*, 238–251.

Baars, B. (1988). *A cognitive theory of consciousness.* Cambridge, UK: Cambridge Universtiy Press.

Bachem, A. (1955). Absolute pitch. *The Journal of the Acoustical Society of America*, *27*(6), 1180–1185.

Baharloo, S., Johnston, P. A., Service, S. K., Gitschier, J. & Freimer, N. B. (1998). Absolute pitch: An approach for identification of genetic and nongenetic components. *American Journal of Human Genetics*, *62*(2), 224–231.

Bailes, F., Dean, R. & Pearce, M. (2013). Music cognition as mental time travel. *Scientific Reports*, *3*, 2690.

Begleiter, R., El-Yaniv, R. & Yona, G. (2004). On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, *22*, 385–421.

Bejerano, G. & Yona, G. (2001). Variations on probabilistic suffix trees: Statistical modeling and prediction of protein families. *Bioinformatics*, *17*(1), 23–43.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127.

Bergeron, M. & Conklin, D. (2011). Subsumption of vertical viewpoint patterns. In C. Agon, M. Andreatta, E. Amiot, J. Bresson & J. Mandereau (Eds.), *Proceedings of mathematics and computation in music, third international conference: MCM 2011* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bharucha, J. J. & Krumhansl, C. L. (1983). The representation of harmonic structure in music: Hierarchies of stability as a function of context. *Cognition*, *13*(1), 63–102.

Bharucha, J. J. & Stoeckig, K. (1987). Priming of chords: Spreading activation or overlapping frequency spectra? *Perception & Psychophysics*, *41*(6), 519–524.

Blacking, J. (1995). *Music, culture and experience*. London: University of Chicago Press.

Boden, M. (2003). *The creative mind: Myths and mechanisms*. London: Routledge.

Bunton, S. (1996). *On-line stochastic processes in data compression* (Doctoral dissertation, University of Washington, Seattle, WA).

Bunton, S. (1997). Semantically motivated improvements for PPM variants. *The Computer Journal*, *40*(2, 3), 76–93.

Cambouropoulos, E. (2001). The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the international computer music conference*. San Fransisco, CA: ICMC.

Cambouropoulos, E. (2010). The musical surface: Challenging basic assumptions. *Musicae Scientiae*, *14*(2 suppl), 131–147.

Cambouropoulos, E. (2015). The harmonic musical surface and two novel chord representation schemes. In D. Meredith (Ed.), *Computational music analysis* (pp. 31–56). Springer International Publishing.

Carrus, E., Pearce, M. & Bhattacharya, J. (2013). Melodic pitch expectation interacts with neural responses to syntactic but not semantic violations. *CORTEX*, *49*(8), 2186–2200.

Castellano, M. A., Bharucha, J. J. & Krumhansl, C. L. (1984). Tonal hierarchies in the music of North India. *Journal of experimental psychology. General*, *113*(3), 394–412.

Chai, W. & Vercoe, B. (2001). Folk music classification using hidden Markov models. In *Proceedings of international conference on artificial intelligence*. Seattle, WA.

Chalmers, D. J. (1996). *The conscious mind: In search of a fundamental theory*. Oxford, UK: OUP.

Chemillier, M. (2004). Toward a formal study of jazz chord sequences generated by Steedman's grammar. *Soft Computing*, *8*(9), 1–6.

Chen, S. F. & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, *13*(4), 359–394.

Cherla, S., Weyde, T., Garcez, A. S. d. & Pearce, M. (2013). A distributed model for multiple-viewpoint melodic prediction. In *Proceedings of 14th international society for music information retrieval conference, ISMIR 2013* (pp. 15–20). Curitiba, Brazil.

Chew, E. (2002). The spiral array: An algorithm for determining key boundaries. In *Music and artificial intelligence* (pp. 18–31). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chomsky, N. (1956). Three models for the description of language. *Information Theory, IRE Transactions on*, *2*(3), 113–124.

Chomsky, N. (1957). *Syntactic structures*. The Hague, Netherlands: Mouton.

Chordia, P., Sastry, A. & Albin, A. (2010). Evaluating multiple viewpoint models of tabla sequences. In *Proceedings of 3rd international workshop on machine learning and music, MML'10* (pp. 21–24). New York, NY: ACM Press.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*, 181–204.

Cleary, J. G. & Teahan, W. J. (1995). Experiments on the zero frequency problem. In *Proceedings of data compression conference, DCC '95* (p. 480). Snowbird, UT: IEEE.

Cleary, J. G. & Teahan, W. J. (1997). Unbounded length contexts for PPM. *The Computer Journal*, *40*(2 and 3), 67–75.

Cleary, J. G. & Witten, I. (1984). Data compression using adaptive coding and partial string matching. *Communications, IEEE Transactions on*, *32*(4), 396–402.

Cleeremans, A. & Dienes, Z. (2008). Computational models of implicit learning. In R. Sun (Ed.), *Cambridge handbook of computational psychology* (pp. 396–421). London, UK.

Cohn, R. (1998). Introduction to neo-Riemannian theory: a survey and a historical perspective. *Journal of Music Theory*, 167–180.

Colton, S. & Wiggins, G. A. (2012). Computational creativity: The final frontier? In *Proceedings of 20th European conference on artificial intelligence, ECAI 2012* (pp. 21–26). Montpellier, France.

Conger, A. J. (1980). Integration and generalization of kappas for multiple raters. *Psychological bulletin*, *88*(2), 322–328.

Conklin, D. (1990). *Prediction and entropy of music* (Master's thesis, Department of Computer Science, University of Calgary).

Conklin, D. (2002). Representation and discovery of vertical patterns in music. In *Music and artificial intelligence* (pp. 32–42). Berlin, Heidelberg: Springer.

Conklin, D. (2003). Music generation from statistical models. In *AISB 2003 symposium on artificial intelligence and creativity in the arts and sciences* (pp. 30–35). Brighton, UK.

Conklin, D. (2010). Discovery of distinctive patterns in music. *Intelligent Data Analysis*, *14*(5), 547–554.

Conklin, D. (2013a). Antipattern discovery in folk tunes. *Journal of New Music Research*, *42*(2), 161–169.

Conklin, D. (2013b). Multiple viewpoint systems for music classification. *Journal of New Music Research*, *42*(1), 19–26.

Conklin, D. (2016). Chord sequence generation with semiotic patterns. *Journal of Mathematics and Music*, *10*(2), 92–106.

Conklin, D. & Anagnostopoulou, C. (2001). Representation and discovery of multiple viewpoint patterns. In *Proceedings of the international computer music conference: ICMC 2001* (pp. 479–485). Havana, Cuba.

Conklin, D. & Anagnostopoulou, C. (2011). Comparative pattern analysis of Cretan folk songs. *Journal of New Music Research*, *40*(2), 119–125.

Conklin, D. & Cleary, J. G. (1988). Modelling and generating music using multiple viewpoints. In *Proceedings of the first workshop on artificial intelligence and music, aaai-88* (pp. 125–137). St. Paul, Minnesota: University of Calgary.

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

Corkill, D. D. (1991). Blackboard systems. *AI Expert*, *6*(9), 40–47.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2001). *Introduction to algorithms.* MIT Press.

Creighton, H. (1966). *Songs and ballads from Nova Scotia.* New York, NY: Dover.

Cross, I. (2001). Music, cognition, culture, and evolution. *Annals of the New York Academy of Sciences*, *930*(1), 28–42.

Crozier, J. B. (1997). Absolute pitch: Practice makes perfect, the earlier the better. *Psychology of Music*, *25*(2), 110–119.

Cuddy, L. L. & Lunney, C. A. (1995). Expectancies generated by melodic intervals - perceptual judgments of melodic continuity. *Perception & Psychophysics*, *57*(4), 451–462.

De Haas, B. W., Wiering, F. & Veltkamp, R. C. (2013). A geometrical distance measure for determining the similarity of musical harmony. *International Journal of Multimedia Information Retrieval*, *2*(3), 189–202.

Deliege, I. (1987). Grouping conditions in listening to music: An approach to Lerdahl & Jackendoff's grouping preference rules. *Music Perception*, *4*(4), 325–359.

Dennett, D. (1991). *Conciousness explained.* Boston, MA: Little, Brown and Co.

Dennett, D. (1996). *Kinds of minds: Toward an understanding of conciousness.* New York, NY: Basic Books.

Desain, P., Honing, H., Vanthienen, H. & Windsor, L. (1998). Computational modeling of music cognition: Problem or solution? *Music Perception*, *16*(1), 151–166.

Deutsch, D., Dooley, K., Henthorn, T. & Head, B. (2009). Absolute pitch among students in an American music conservatory: Association with tone language fluency. *The Journal of the Acoustical Society of America*, *125*(4), 2398–2403.

Dowling, W. J. (1978). Scale and contour: Two components of a theory of memory for melodies. *Psychological Review*, *85*, 341–354.

Dowling, W. J. & Bartlett, J. C. (1981). The importance of interval information in long-term memory for melodies. *Psychomusicology: A Journal of Research in Music Cognition*, *1*(1), 30.

Dowling, W. J. & Fujitani, D. S. (1971). Contour, interval, and pitch recognition in memory for melodies. *The Journal of the Acoustical Society of America*, *49*(2), 524–531.

Dubnov, S., Assayag, G., Lartillot, O. & Bejerano, G. (2003). Using machine-learning methods for musical style modeling. *Computer*, *36*(10), 73–80.

Ebcĭoğlu, K. (1986). An expert system for chorale harmonization. In *Proceedings of the 5th national conference on artificial intelligence* (pp. 784–788). Menlo Park, CA: AAAI Press.

Ebcĭoğlu, K. (1990). An expert system for harmonizing chorales in the style of J.S. Bach. *The Journal of Logic Programming*, *8*(1), 145–185.

Eerola, T. (2003). *The dynamics of musical expectancy* (Doctoral dissertation, University of Jyväskylä).

Eerola, T. (2004). Data-driven influences on melodic expectancy: Continuations in North Sami yoiks rated by South African traditional healers. In S. D. Lipscombe, R. Ashley, R. O. Gjerdingen & P. Webster (Eds.), *Proceedings of the 8th international conference on music perception cognition, ICMPC8* (pp. 83–87). Evanston, IL.

Eerola, T., Jäärvinen, T., Louhivuori, J. & Toiviainen, P. (2001). Statistical features and perceived similarity of folk melodies. *Music Perception*, *18*(3), 275–296.

Egermann, H., Pearce, M., Wiggins, G. A. & McAdams, S. (2013). Probabilistic models of expectation violation predict psychophysiological emotional responses to live concert music. *Cognitive, Affective, & Behavioral Neuroscience*, *13*(3), 533–553.

Farbood, M. (2010). Working memory and the perception of hierarchical tonal structures. In *Proceedings of the 11th international conference of music perception and cognition, ICMPC11* (pp. 119–222). Seattle, Washington.

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Pschological Bulletin*, *76*(5), 378–382.

Forte, A. (1962). *Tonal harmony in concept and practice.* New York: Holt, Rinehart, and Winston.

Forte, A. (1973). *The structure of atonal music.* Yale University Press.

Forth, J. (2012). *Cognitively-motivated geometric methods of pattern discovery and models of similarity in music* (Doctoral dissertation, Goldsmiths, University of London, London).

Forth, J., Agres, K., Purver, M. & Wiggins, G. A. (2016). Entraining IDyOT: Timing in the information dynamics of thinking. *Frontiers in Psychology*, *7*(68), 23–19.

Frankel, R. E., Rosenschein, S. J. & Smoliar, S. W. (1976). A LISP-based system for the study of Schenkerian analysis. *Computers and the Humanities*, *10*(1), 21–32.

Frankland, B. W. & Cohen, A. J. (2004). Parsing of melody: Quantification and testing of the local grouping rules of Lerdahl and Jackendoff's A Generative Theory of Tonal Music. *Music Perception*, *21*(4), 499–543.

Gardenfors, P. (2000). *Conceptual spaces: The geometry of thought.* Cambridge, MA: MIT Press.

Gebauer, L., Kringelbach, M. L. & Vuust, P. (2012). Ever-changing cycles of musical pleasure: The role of dopamine and anticipation. *Psychomusicology: Music, Mind, and Brain*, *22*(2), 152–167.

Granroth-Wilding, M. (2013). *Harmonic analysis of music using combinatory categorial grammar* (Doctoral dissertation, School of Informatics, University of Edinburgh).

Granroth-Wilding, M. & Steedman, M. (2014). A robust parser-interpreter for jazz chord sequences. *Journal of New Music Research*, *43*(4), 355–374.

Griffiths, S., Purver, M. & Wiggins, G. A. (2015). From phoneme to morpheme: A computational model. In *Proceedings of 6th quantitative investigations in theoretical linguistics conference, QITL-6.* Tübingen, Germany.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences: Computer science and computational biology.* Cambridge University Press.

Hansen, N. C. & Pearce, M. (2014). Predictive uncertainty in auditory sequence processing. *Frontiers in Psychology*, *5*(25), 1–17.

Harris, M., Smaill, A. & Wiggins, G. A. (1991). Representing music symbolically. In A. Camurri & C. Canepa (Eds.), *Xi colloquio di informatica musicale* (pp. 55–69). Genoa, Italy.

Harte, C., Sandler, M. B., Abdallah, S. M. & Gómez, E. (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of 6th international society for music information retrieval conference, ISMIR 2005* (pp. 66–71). London, UK.

Hebb, D. (1949). *The organization of behaviour.* New York, NY: Wiley.

Hedges, T. W. & Rohrmeier, M. A. (2011). Exploring Rameau and beyond: A corpus study of root progression theories. In C. Agon, E. Amiot, M. Andreatta, G. Assayag, J. Bresson & J. Manderau (Eds.), *Proceedings of mathematics and computation in music, third international conference: MCM 2011* (pp. 334–337). Berlin, Heidelberg: Springer.

Hedges, T. W., Roy, P. & Pachet, F. (2014). Predicting the composer and style of jazz chord progressions. *Journal of New Music Research*, *43*(3), 276–290.

Hedges, T. W. & Wiggins, G. A. (2015). *Segmentation and grouping structures in jazz chord sequences: An information-theoretic approach.* Paper presented at Speech Language, Music, Art, Reasoning Thought Conference: SMART 2015. Amsterdam, Netherlands.

Hedges, T. W. & Wiggins, G. A. (2016a). Improving predictions of derived viewpoints in multiple viewpoint systems. In *Proceedings of 17th international society for music information retrieval conference, ISMIR 2016* (pp. 420–426). New York, NY.

Hedges, T. W. & Wiggins, G. A. (2016b). The prediction of merged attributes with multiple viewpoint systems. *Journal of New Music Research*, 1–19.

Herremans, D., Weisser, S., Sörensen, K. & Conklin, D. (2015). Generating structured music for bagana using quality metrics based on Markov models. *Expert Systems with Applications*, *42*(21), 7424–7435.

Hillewaere, R., Manderick, B. & Conklin, D. (2012). String methods for folk tune genre classification. In *Proceedings of 13th international society for music information retrieval conference, ISMIR 2012* (pp. 217–222). Porto, Portugal.

Hillewaere, R., Manderick, B. & Conklin, D. (2009). Global feature versus event models for folk song classification. In *Proceedings of 10th international society for music information retrieval conference, ISMIR 2009* (pp. 729–733). Kobe, Japan.

Hinton, G. E. (1999). Products of experts. In *Proceedings of the 9th international conference on artificial neural networks, ICANN '99* (pp. 1–6). London, UK: IEE.

Hinton, G. E. (2000). *Training products of experts by minimizing contrastive divergence* (tech. rep. No. GCNU TR 2000-004). Gatsby Computational Neuroscience Unit, University College London.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the national academy of sciences of the United States of America* (pp. 2554–2558). National Acad Sciences.

Howard, P. G. (1993). *The design and analysis of efficient lossless data compression systems* (Doctoral dissertation, Department of Computer Science, Brown University, Providence, RI).

Huron, D. (2006). *Sweet anticipation: Music and the psychology of expectation.* Cambridge, MA: MIT Press.

Ihara, S. (1993). *Information theory for coninuous systems.* Singapore: World Scientific.

Jackendoff, R. (1987). *Conciousness and the computational mind.* Cambridge, MA: MIT Press.

Jelinek, F. & Mercer, R. L. (1980). Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the workshop on pattern recognition in practice* (pp. 381–397). Amsterdam, The Netherlands: North-Holland.

Jonaitis, E. M. & Saffran, J. R. (2009). Learning harmony: The role of serial statistics. *Cognitive Science*, *33*(5), 951–968.

Jurafsky, D. & Martin, J. H. (2009). *Speech and language processing* (2nd). An introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, NJ: Pearson.

Karlsson, F. (2007). Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, *43*(02), 365.

Kassler, M. (1975). *Proving musical theorems I: The middleground of Heinrich Schenker's theory of tonality* (tech. rep. No. 103). Basser Department of Computer Science, University of Sydney.

Kassler, M. (1988). APL applied in music theory. In *Proceedings of the international conference on APL: APL'88* (pp. 209–214). New York, USA: ACM Press.

Kirlin, P. B. (2014). *A probabilistic model of hierarchical music analysis* (Doctoral dissertation, University of Massachusetts Amherst).

Kneser, R. & Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of international conference on acoustics, speech and signal processing, ICASSP 1995* (pp. 181–184). Detroit, MI.

Knopoff, L. & Hutchinson, W. (1983). Entropy as a measure of style: The influence of sample length. *Journal of Music Theory*, *27*(1), 75.

Koelsch, S., Busch, T., Jentschke, S. & Rohrmeier, M. A. (2016). Under the hood of statistical learning: A statistical MMN reflects the magnitude of transitional probabilities in auditory sequences. *Scientific Reports*, *6*(1), 19741.

Koelsch, S., Kilches, S., Steinbeis, N. & Schelinski, S. (2008). Effects of unexpected chords and of performer's expression on brain responses and electrodermal activity. *PLoS ONE*, *3*(7), e2631.

Koelsch, S., Rohrmeier, M. A., Torrecuso, R. & Jentschke, S. (2013). Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(38), 15443–15448.

Kohonen, T. (1997). *Self-organising maps* (2nd ed.). Berlin, Germany: Springer.

Kostka, S. & Payne, D. (1984). *Tonal harmony with an introduction to twentieth century music.* McGraw-Hill. New York, NY.

Krichevsky, R. E. & Trofimov, V. K. (1981). The performance of universal encoding. *Information Theory, IEEE Transactions on*, *27*(2), 199–207.

Kröger, P., Passos, A., Sampaio, M. & De Cidra, G. (2008). Rameau: A system for automatic harmonic analysis. In *Proceedings of the international computer music conference, ICMC 2008* (pp. 273–281). Belfast, Northern Ireland.

Krumhansl, C. L. (1990). *Cognitive foundations of musical pitch*. Oxford, UK: Oxford University Press.

Krumhansl, C. L. (1995). Effects of musical context on similarity and expectancy. *Systematische Musikwissenschaft*, *3*(2), 211–250.

Krumhansl, C. L., Bharucha, J. J. & Castellano, M. A. (1982a). Key distance effects on perceived harmonic structure in music. *Perception & Psychophysics*, *32*(2), 96–108.

Krumhansl, C. L., Bharucha, J. J. & Kessler, E. J. (1982b). Perceived harmonic structure of chords in three related musical keys. *Journal of Experimental Psychology: Human Perception and Performance*, *8*(1), 24–36.

Krumhansl, C. L. & Kessler, E. J. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, *89*(4), 334–368.

Krumhansl, C. L., Louhivuori, J., Toiviainen, P., Järvinen, T. & Eerola, T. (1999). Melodic expectation in Finnish spiritual folk hymns: Convergence of statistical, behavioral, and computational approaches. *Music Perception*, *17*(2), 151–195.

Krumhansl, C. L. & Shepard, R. N. (1979). Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of Experimental Psychology: Human Perception and Performance*, *5*(4), 579–594.

Krumhansl, C. L., Toivanen, P., Eerola, T., Toiviainen, P., Järvinen, T. & Louhivuori, J. (2000). Cross-cultural music cognition: Cognitive methodology applied to North Sami yoiks. *Cognition*, *76*(1), 13–58.

Landis, J. R. & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174.

Larson, S. (2012). *Musical forces: Motion, metaphor, and meaning in music*. Indiana University Press.

Leistikow, R. J. (2006). *Bayesian modeling of musical expectations via maximum entropy stochastic grammars* (Doctoral dissertation, Stanford University, CA).

Leonard, H. (2012). *The real book: volume I, II, III, IV and V*. Winoa, MN: Hal Leonard.

Lerdahl, F. & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, MA: MIT Press.

Levine, M. (1989). *The jazz piano book*. Petaluma, CA: Sher Music Co.

Levine, M. (1995). *The jazz theory book*. Petaluma, CA: Sher Music Co.

Levinson, S. C. (2016). Turn-taking in human communication – origins and implications for language processing. *Trends in Cognitive Sciences*, *20*(1), 6–14.

Levinson, S. C. & Torreira, F. (2015). Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*, *6*(268), 247–17.

Levitin, D. J. (1994). Absolute memory for musical pitch: Evidence from the production of learned melodies. *Perception & Psychophysics*, *56*(4), 414–423.

Lewin, D. (1987). *Generalised musical intervals and transformations.* New Haven, CT: Yale University Press.

Longuet-Higgins, H. C. (1978). The grammar of music. *Interdisciplinary Science Reviews*, *3*, 148–156.

Longuet-Higgins, H. C. (1979). Review lecture: The perception of music. In *Proceedings of the royal society* (pp. 307–322). London, UK.

Loui, P. & Wessel, D. (2008). Learning and liking an artificial musical system: Effects of set size and repeated exposure. *Musicae Scientiae*, *12*(2), 207–230.

Loui, P. (2011). Statistical learning: What can music tell us? In P. Rebuschat & J. Williams (Eds.), *Statistical learning and language acquisition* (pp. 433–462). The Hague, Netherlands: Mouton de Gruyter.

Loui, P., Wessel, D. L. & Hudson Kam, C. L. (2010). Humans rapidly learn grammatical structure in a new musical scale. *Music Perception*, *27*(5), 377–388.

MacKay, D. (1998). Introduction to Monte Carlo methods. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 175–204). Dordrecht, Netherlands: Springer Netherlands.

MacKay, D. (2003). *Information theory, inference and learning algorithms.* Cambridge, UK: Cambridge University Press.

Mann, A. (1965). *The study of counterpoint: From Johann Joseph Fux's Gradus ad Parnassum.* London, England: W. W. Norton & Company.

Manning, C. & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, MA: MIT Press.

Manzara, L., Witten, I. & James, M. (1992). On the entropy of music: An experiment with Bach chorale melodies. *Leonardo Music Journal*, *2*(1), 81–88.

Marsden, A. (2005). Generative structural representation of tonal music. *Journal of New Music Research*, *34*(4), 409–428.

Marsden, A. (2010). Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, *39*(3), 269–289.

Martin, H. (2012). Expanding jazz tonality: The compositions of John Coltrane. *Theory and Practice*, *37/38*, 185–219.

Mathews, M. V., Pierce, J. R., Reeves, A. & Roberts, L. A. (1988). Theoretical and experimental explorations of the Bohlen–Pierce scale. *The Journal of the Acoustical Society of America*, *84*(4), 1214–1222.

Mauch, M., Noland, K. & Dixon, S. (2009). Using musical structure to enhance automatic chord transcription. In *10th international society for music information retrieval conference* (pp. 231–236). Kobe, Japan.

Mavromatis, P. & Brown, M. (2004). Parsing context-free grammars for music: A computational model of schenkerian analysis. In *Proceedings of the 8th international conference on music perception cognition, ICMPC8* (pp. 414–415). Evanston, IL.

Maxwell, H. J. (1992). An expert system for harmonic analysis of tonal music. In M. Balaban, K. Ebcı̆oğlu & O. Laske (Eds.), *Understanding music with AI* (pp. 335–353). Menlo Park, CA.

Mazzola, G. (2002). *The topos of music: geometric logic of concepts, theory, and performance*. Basel: Birkhäuser.

McDermott, J. H. & Oxenham, A. J. (2008). Music perception, pitch, and the auditory system. *Current Opinion in Neurobiology*, *18*(4), 452–463.

McVicar, M., Santos-Rodriguez, R., Ni, Y. & Bie, T. D. (2014). Automatic chord estimation from audio: A review of the state of the art. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, *22*(2), 556–575.

Meeus, N. (2000). Toward a post-Schoenbergian grammar of tonal and pre-tonal harmonic progressions. *Music Theory Online*, *6*(1).

Meyer, L. (1956). *Emotion and meaning in music*. London, UK: University of Chicago Press.

Miyazaki, K. (1988). Musical pitch identification by absolute pitch possessors. *Perception & Psychophysics*, *44*(6), 501–512.

Miyazaki, K. (1989). Absolute pitch identification: Effects of timbre and pitch region. *Music Perception*, *7*(1), 1–14.

Miyazaki, K. (1990). The speed of musical pitch identification by absolute-pitch possessors. *Music Perception*, *8*(2), 177–188.

Moffat, A., Neal, R. M. & Witten, I. (1998). Arithmetic coding revisited. *Information Systems, ACM Transactions on*, *16*(3), 256–294.

Moffat, A., Sharman, N., Witten, I. H. & Bell, T. C. (1994). An empirical evaluation of coding methods for multi-symbol alphabets. *Information Processing and Management*, *30*(6), 791–804.

Moffat, A. (1990). Implementing the ppm data compression scheme. *Communications, IEEE Transactions on*, *38*(11), 1917–1921.

Murphy, K. (2002). *Dynamic bayesian networks: Representation, inference and learning* (Doctoral dissertation, University of California Press, Berkeley, CA).

Narmour, E. (1990). *The analysis and cognition of basic melodic structures: The implication-realization model.* Chicago, IL: University of Chicago Press.

Omigie, D., Pearce, M. & Stewart, L. (2012). Tracking of pitch probabilities in congenital amusia. *Neuropsychologia*, *50*(7), 1483–1493.

Omigie, D., Pearce, M., Williamson, V. J. & Stewart, L. (2013). Electrophysiological correlates of melodic processing in congenital amusia. *Neuropsychologia*, *51*(9), 1749–1762.

Pachet, F. & Roy, P. (2014a). Imitative leadsheet generation with user constraints. In T. Schaub, G. Friedrich & B. OSullivan (Eds.), *Proceedings of 20th European conference on artificial intelligence, ECAI 2012* (pp. 1077–1078). Prague, Czech Republic.

Pachet, F. & Roy, P. (2014b). Non-conformant harmonization: the real book in the style of take 6. In *Proceedings of the 5th international conference on computational creativity, ICCC 2014*. Ljubljana, Slovenia.

Pachet, F., Suzda, J. & Martín, D. (2013). A comprehensive online database of machine-readable leadsheets for jazz standards. In *Proceedings of 14th international society for music information retrieval conference, ISMIR 2013* (pp. 275–280). Curitiba, Brazil.

Pachet, F. (2000). Computer analysis of jazz chord sequences: Is Solar a blues? In E. R. Miranda (Ed.), *Readings in music and artificial intelligence.* Harwood Academic Publishers.

Pachet, F. (2003). The continuator: Musical interaction with style. *Journal of New Music Research*, *32*(3), 333–341.

Pachet, F. (2012). Musical virtuosity and creativity. In J. McCormack & M. d'Inverno (Eds.), *Computers and creativity* (pp. 115–146). Berlin, Heidelberg: Springer Berlin Heidelberg.

Pachet, F. & Roy, P. (2011). Markov constraints: Steerable generation of markov sequences. *Constraints*, *16*(2), 148–172.

Paiement, J. F. (2008). *Probabilistic models for music* (Doctoral dissertation, University of Monteal, Canada).

Paiement, J. F., Eck, D. & Bengio, S. (2005). A probabilistic model for chord progressions. In *Proceedings of the 6th international conference of music information retrieval, ISMIR 2005* (pp. 312–319). London, UK.

Papadopoulos, A., Roy, P. & Pachet, F. (2014). Avoiding plagiarism in Markov sequence generation. In *Proceedings of the twenty-eighth AAAI conference on artificial intelligence* (pp. 2731–2737). Québec City, Canada.

Pardo, B. & Birmingham, W. P. (2002). Algorithms for chordal analysis. *Computer Music Journal*, *26*(2), 27–49.

Pearce, M. (2005). *The construction and evaluation of statistical models of melodic structure in music perception and composition* (Doctoral dissertation, City University, London, UK).

Pearce, M., Conklin, D. & Wiggins, G. A. (2005). Methods for combining statistical models of music. In U. K. Wiil (Ed.), *Proceedings of the 2nd international conference on computer music modeling and retrieval, CMMR'04* (pp. 295–312). Berlin, Heidelberg: Springer-Verlag.

Pearce, M., Mullensiefen, D. & Wiggins, G. A. (2010a). Melodic grouping in music information retrieval: New methods and applications. In Z. Ras & A. Wieczorkowska (Eds.), *Advances in music information retrieval* (pp. 364–388). Berlin, Heidelberg: Springer.

Pearce, M., Mullensiefen, D. & Wiggins, G. A. (2010b). The role of expectation and probabilistic learning in auditory boundary perception: A model comparison. *Perception*, *39*(10), 1365–1389.

Pearce, M., Ruiz, M. H., Kapasi, S., Wiggins, G. A. & Bhattacharya, J. (2010c). Unsupervised statistical learning underpins computational, behavioural, and neural manifestations of musical expectation. *NeuroImage*, *50*(1), 302–313.

Pearce, M. & Wiggins, G. A. (2004). Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, *33*(4), 367–385.

Pearce, M. & Wiggins, G. A. (2006). Expectation in melody: The influence of context and learning. *Music Perception*, *23*(5), 377–405.

Pearce, M. & Wiggins, G. A. (2007). Evaluating cognitive models of musical composition. In A. Cardoso & G. A. Wiggins (Eds.), *4th international joint workshop on computational creativity* (pp. 73–80). London, UK: Goldsmiths, University of London.

Pearce, M. & Wiggins, G. A. (2012). Auditory expectation: The information dynamics of music perception and cognition. *Topics in Cognitive Science*, *4*(4), 625–652.

Perez-Sancho, C., Rizo, D. & Inesta, J. M. (2009). Genre classification using chords and stochastic language models. *Connection Science*, *21*(2), 145–159.

Perruchet, P. & Vinter, A. (1998). PARSER: A model for word segmentation. *Journal of Memory and Language*, *39*, 246–263.

Pinker, S. (1997). *How the mind works.* New York, NY: W. W. Norton & Company.

Piston, W. (1948). *Harmony.* New York: W. W. Norton and Company.

Plantinga, J. & Trainor, L. J. (2005). Memory for melody: Infants use a relative pitch code. *Cognition*, *98*(1), 1–11.

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

Popper, K. (1934). *The logic of scientific discovery.* London: Hutchinson & Co.

Profita, J. & Bidder, T. G. (1988). Perfect pitch. *American Journal of Medical Genetics*, *29*(4), 763–771.

Rameau, J. P. (1971). *Treatise on harmony*. New York: Dover Publications.

Raphael, C. & Stoddard, J. (2004). Harmonic analysis with probabilistic graphical models. *Computer Music Journal*, *28*(3), 45–52.

Rebuschat, P. & Williams, J. (Eds.). (2012). *Statistical learning and language acquisition*. The Hague, Netherlands: Mouton de Gruyter.

Riemann, H. (1895). *Harmony simplified: or, The theory of the tonal functions of chords*. London: Augener.

Riemenschneider, A. (1941). *371 harmonised chorales and 69 chorale melodies with figured bass*. New York, NY: G. Schirmer Inc.

Rissanen, J. (1983). A universal data compression system. *Information Theory, IEEE Transactions on*, *29*(5), 656–664.

Ritchie, G. (2006). The transformational creativity hypothesis. *New Generation Computing*, *24*(3), 241–266.

Rohrmeier, M. A. (2010). *Implicit learning of musical structure* (Doctoral dissertation, University of Cambridge, Cambridge).

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

Rohrmeier, M. A. & Cross, I. (2008). Statistical properties of tonal harmony in Bach's chorales, 619–627.

Rohrmeier, M. A. & Cross, I. (2009). Tacit tonality - implicit learning of context-free harmonic structure. In T. Eerola, S. Saarikallio, T. Himberg & P.-S. Eerola (Eds.), *Proceedings of the 7th triennial conference of European society for the cognitive sciences of music, ESCOM 2009*. Jyväskylä, Finland.

Rohrmeier, M. A. & Cross, I. (2014). Modelling unsupervised online-learning of artificial grammars: Linking implicit and statistical learning. *Consciousness and Cognition*, *27*, 155–167.

Rohrmeier, M. A. & Graepel, T. (2012). Comparing feature-based models of harmony. In *Proceedings of the 9th international symposium on computer music modeling and retrieval, CMMR 2012* (pp. 357–370). London, UK: Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval.

Rohrmeier, M. A., Rebuschat, P. & Cross, I. (2011). Incidental and online learning of melodic structure. *Consciousness and Cognition*, *20*(2), 214–222.

Ron, D., Singer, Y. & Tishby, N. (1996). The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, *25*(2-3), 117–149.

Rosch, E. H. (1973). Natural categories. *Cognitive Psychology*, *4*(3), 328–350.

Roy, P. & Pachet, F. (2013). Enforcing meter in finite-length Markov sequences. In *Proceedings of twenty-seventh AAAI conference on artificial intelligence* (pp. 854–861). Bellevue, WA.

Russell, S. J. & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd). Englewood Cliffs, NJ: Prentice-Hall.

Ruwet, N. (1972). *Language, musique, poésie*. Paris: Editions du Seuil.

Saffran, J. R., Aslin, R. N. & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science*, *274*(5294), 1926–1928.

Saffran, J. R. & Griepentrog, G. J. (2001). Absolute pitch in infant auditory learning: Evidence for developmental reorganization. *Developmental Psychology*, *37*(1), 74.

Saffran, J. R., Johnson, E. K., Aslin, R. N. & Newport, E. L. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition*, *70*(1), 27–52.

Saffran, J. R., Reeck, K., Niebuhr, A. & Wilson, D. (2005). Changing the tune: The structure of the input affects infants' use of absolute and relative pitch. *Developmental Science*, *8*(1), 1–7.

Sawilowsky, S. S. (2009). New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, *8*(2), 597–599.

Schaffrath, H. (1995). *The Essen folksong collection: Database containing folksong transcriptions in the Kern format and a 34-page research guide.* (D. Huron, Ed.). Menlo Park, CA.

Schellenberg, E. G. (1996). Expectancy in melody: Tests of the implication-realization model. *Cognition*, *58*(1), 75–125.

Schellenberg, E. G. (1997). Simplifying the implication-realization model of melodic expectancy. *Music Perception*, *14*(3), 295–318.

Schellenberg, E. G. & Trehub, S. E. (2003). Good pitch memory is widespread. *Psychological Science*, *14*(3), 262–266.

Schellenberg, E. G. & Trehub, S. E. (2008). Is there an Asian advantage for pitch memory? *Music Perception*, *25*(3), 241–252.

Schenker, H. (1979). *Free composition*. New York, NY: Longman.

Schmuckler, M. A. (1997). Expectancy effects in memory for melodies. *Canadian Journal of Experimental Psychology*, *51*(4), 292–306.

Schmuckler, M. A. (1989). Expectation in music: Investigation of melodic and harmonic processes. *Music Perception*, *7*(2), 109–149.

Schmuckler, M. A. (1990). The performance of global expectations. *Psychomusicology: A Journal of Research in Music Cognition*, *9*(2), 122–147.

Schoenberg, A. (1969). *Structural functions of harmony* (L. Stein, Ed.). New York: W. W. Norton and Company.

Sergeant, D. (1969). Experimental investigation of absolute pitch. *Journal of Research in Music Education*, *17*(1), 135–143.

Sertan, S. & Chordia, P. (2011). Modeling melodic improvisation in Turkish folk music using variable-length Markov models. In *Proceedings of the 12th international society for music information retrieval conference, ISMIR 2011* (pp. 269–274). Miami, FL.

Servan-Schreiber, E. & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *16*(4), 592–608.

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *8*, 333–343.

Shkarin, D. (2002). PPM: One step to practicality. In *Proceedings of the data compression conference, DCC* (pp. 202–211). Snowbird, UT: IEEE Comput. Soc.

Simpson, J. & Huron, D. (1994). Absolute pitch as a learned phenomenon: Evidence consistent with the Hick-Hyman law. *Music Perception*, *12*(2), 267–270.

Sloboda, J. A. (1985). *The musical mind: The cognitive psychology of music*. Oxford: Oxford University Press.

Srinivasamurthy, A. & Chordia, P. (2012). Multiple viewpoint modeling of North Indian classical vocal compositions. In *The 9th international symposium on computer music modeling and retrieval, CMMR 2012* (pp. 344–356). London, UK.

Steedman, M. (1984). A generative grammar for jazz chord sequences. *Music Perception*, *2*(1), 52–77.

Steedman, M. (2000). *The syntactic process*. Cambridge, MA: MIT Press.

Steinbeis, N., Koelsch, S. & Sloboda, J. A. (2006). The role of harmonic expectancy violations in musical emotions: Evidence from subjective, physiological, and neural responses. *Journal of Cognitive Neuroscience*, *18*(8), 1380–1393.

Sturm, B. L. (2013). Classification accuracy is not enough. *Journal of Intelligent Information Systems*, *41*(3), 371–406.

Sturm, B. L. (2014). A simple method to determine if a music information retrieval system is a “horse”. *Multimedia, IEEE Transactions on*, *16*(6), 1636–1644.

Taine, H. (1871). *On intelligence*. Savill, Edwards and Co. London, UK.

Temperley, D. (1999). What’s key for key? The Krumhansl-Schmuckler key-finding algorithm reconsidered. *Music Perception*, *17*(1), 65–100.

Temperley, D. (2001). *The cognition of basic musical structures*. Cambridge, MA: MIT Press.

Temperley, D. (2014). Probabilistic models of melodic interval. *Music Perception*, *32*(1), 85–99.

Theusch, E. & Gitschier, J. (2011). Absolute pitch twin study and segregation analysis. *Twin Research and Human Genetics*, *14*(2), 173–178.

Thompson, W. F., Cuddy, L. L. & Plaus, C. (1997). Expectancies generated by melodic intervals: Evaluation of principles of melodic implication in a melody-completion task. *Perception & Psychophysics*, *59*(7), 1069–1076.

Tillmann, B. & McAdams, S. (2004). Implicit learning of musical timbre sequences: Statistical regularities confronted with acoustical (dis)similarities. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(5), 1131–1142.

Trehub, S. E., Bull, D. & Thorpe, L. A. (1984). Infants' perception of melodies: The role of melodic contour. *Child Development*, *55*(3), 821.

Triviño-Rodriguez, J. L. & Morales-Bueno, R. (2001). Using multiattribute prediction suffix graphs to predict and generate music. *Computer Music Journal*, *25*(3), 62–79.

Ukkonen, E. (1995). On-line construction of suffix trees. *Algorithmica*, *14*(3), 249–260.

Ulrich, J. (1977). The analysis and synthesis of jazz by computer. (pp. 865–872). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, *13*(2), 260–269.

Volf, P. (2002). *Weighting techniques in data compression: Theory and algorithms* (Doctoral dissertation, Eindhoven, Netherlands).

Volkova, A., Trehub, S. E. & Schellenberg, E. G. (2006). Infants' memory for musical performances. *Developmental Science*, *9*(6), 583–589.

Wallace, G. (1926). *The art of thought*. New York, NY: Harcourt Brace.

Ward, W. & Burns, M. (1982). Absolute pitch. In D. Deutsch (Ed.), *The psychology of music* (pp. 431–451). San Diago, CA.

Waters, K. (2010). "Giant Steps" and the ic4 legacy. *Intégral*, *24*(Special Issue in honor of Robert Wason), 135–162.

Whorley, R. (2013). *The construction and evaluation of statistical models of melody and harmony* (Doctoral dissertation, Goldsmiths, University of London, London).

Whorley, R. & Conklin, D. (2016). Music generation from statistical models of harmony. *Journal of New Music Research*, *45*(2), 1–24.

Whorley, R., Rhodes, C., Wiggins, G. A. & Pearce, M. (2013a). Harmonising melodies: Why do we add the bass line first? In *Proceedings of the 4th international conference on computational creativity, ICCC 2013* (pp. 79–86). Sydney, Australia.

Whorley, R., Wiggins, G. A., Rhodes, C. & Pearce, M. (2013b). Multiple viewpoint systems: time complexity and the construction of domains for complex musical viewpoints in the harmonization problem. *Journal of New Music Research*, *42*(3), 237–266.

Wiggins, G. A. (1998). Music, syntax, and the meaning of "meaning". *Proceedings of First Symposium on Music and Computers*, 18–23.

Wiggins, G. A. (2007). Models of musical similarity. *Musicae Scientiae*, *11*(1 Suppl), 315–338.

Wiggins, G. A. (2009). Semantic gap?? Schemantic schmap!! Methodological considerations in the scientific study of music. In *11th IEEE international symposium on multimedia* (pp. 477–482). San Diego, CA: IEEE.

Wiggins, G. A. (2010). Cue abstraction, paradigmatic analysis and information dynamics: Towards music analysis by cognitive model. *Musicae Scientiae*, *14*(2), 307–331.

Wiggins, G. A. (2011). Computer models of (music) cognition. In P. Rebuschat, J. Hawkins & I. Cross (Eds.), *Language and music as cognitive systems* (pp. 169–188). Oxford, UK: Oxford University Press.

Wiggins, G. A. (2012a). ‘‘I let the music speak'': Cross-domain application of a cognitive model of musical learning. In P. Rebuschat & J. Williams (Eds.), *Statistical learning and language acquisition* (pp. 463–494). Amsterdam, NL: Mouton De Gruyter.

Wiggins, G. A. (2012b). Music, mind and mathematics: Theory, reality and formality. *Journal of Mathematics and Music*, *6*(2), 111–123.

Wiggins, G. A. (2012c). The mind's chorus: Creativity before consciousness. *Cognitive Computation*, *4*(3), 306–319.

Wiggins, G. A. & Forth, J. (2015). IDyOT: A computational theory of creativity as everyday reasoning from learned information. In T. R. Besold, M. Schorlemmer & A. Smaill (Eds.), *Computational creativity research towards creative machines* (pp. 127–148). Amsterdam, Netherlands: Atlantis Press.

Wiggins, G. A., Miranda, E., Smaill, A. & Harris, M. (1993). A framework for the evaluation of music representation systems. *Computer Music Journal*, *17*(3), 31–42.

Wiggins, G. A., Mullensiefen, D. & Pearce, M. (2010). On the non-existence of music: Why music theory is a figment of the imagination. *Musicae Scientiae, Discussion Forum*, *5*, 231–255.

Willems, F. M. J., Shtarkov, Y. M. & Tjalkens, T. J. (1995). The context-tree weighting method: Basic properties. *Information Theory, IEEE Transactions on*, *41*(3), 653–664.

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

Witmer, R. & Kernfeld, B. (2002). Fake book. In *The new Grove dictionary of jazz*. Oxford: Oxford University Press.

Witten, I. & Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, *37*(4), 1085–1094.

Woolhouse, M., Cross, I. & Horton, T. (2006). The perception of non-adjacent harmonic relations. In *Proceedings of the 9th conference on music perception and cognition, 9th ICMPC* (pp. 1236–1244). Bologna, Italy.

Woolhouse, M., Cross, I. & Horton, T. (2016). Perception of nonadjacent tonic-key relationships. *Psychology of Music*, *44*(4), 802–815.

Youngblood, J. E. (1958). Style as information. *Journal of Music Theory*, *2*(1), 24.

Zanette, D. H. (2006). Zipf's law and the creation of musical context. *Musicae Scientiae*, *10*(1), 3–18.

Zbikowski, L. (2002). *Conceptualizing music: Cognitive structure, theory, and analysis*. Oxford University Press.

Zipf, G. K. (1935). *The psycho-biology of language*. Boston, MA: Houghton-Mifflin.

Zipf, G. K. (1949). *Human behaviour and the principle of least effort*. Cambridge, MA: Addison-Wesley.

Ziv, J. & Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, *24*(5), 530–536.