EVALUATING A SIMPLE APPROACH TO MUSIC INFORMATION RETRIEVAL:

CONCEIVING MELODIC N-GRAMS AS TEXT

by

J. Stephen Downie

Faculty of Information and Media Studies
Graduate Program in Library and Information Science

Submitted in partial fulfillment
of the requirements of the degree of
Doctor of Philosophy

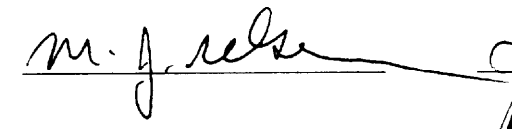
Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
July, 1999

THE UNIVERSITY OF WESTERN ONTARIO
FACULTY OF GRADUATE STUDIES

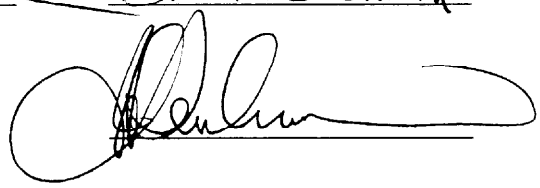
CERTIFICATE OF EXAMINATION

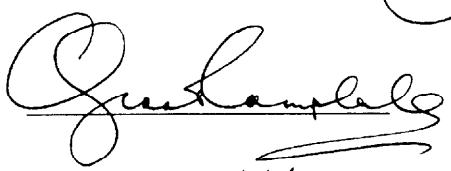
Chief advisor

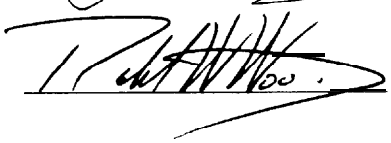
Examining Board



Advisory Committee



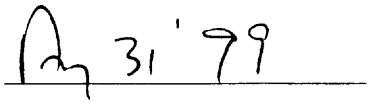




The thesis by
J. Stephen Downie

entitled
Evaluating a Simple Approach to Music Information Retrieval:
Conceiving Melodic N-Grams as Text

is accepted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Date 

Chair of Examining Board

Abstract

Taking our cue from those printed thematic catalogues that have reduced the amount of music information represented we developed, and then evaluated, a Music Information Retrieval (MIR) system based upon the intervals found within the melodies of a collection of 9354 folksongs. We believe that there is enough information contained within an interval-only representation of monophonic melodies that effective retrieval of music information has been achieved. We extended the thematic catalogue model by affording access to musical expressions found anywhere within a melody. To achieve this extension we fragmented to the melodies into length-n subsections called n-grams. The length of these n-grams and the degree to which we precisely represent the intervals are variables analyzed in this thesis.

N-grams form discrete units of melodic information much in the same manner as words are discrete units of language. Thus, we have come to consider them "musical words." This implies that, for the purposes of music information retrieval, we can treat them as "real words" and thereby apply traditional text-based information retrieval techniques. We examined the validity of our "musical word" concept in two ways. First, a variety of informetric analyses were conducted to examine in which ways the informetric properties of "musical words" and "real words" are similar or different. Second, we constructed a collection of "musical word" databases using the famous text-based, SMART information retrieval system. A group of simulated queries was run against these databases. The results were evaluated using the normalized precision and normalized recall measures. Results indicate that the simple approach to music information retrieval examined in this study shows great merit.

Keywords: Music Information Retrieval, Information Retrieval, Informetrics, Informetric Modeling, Information Systems, Information System Evaluation.

Acknowledgements

The writing of a thesis is a long and arduous task. Only with the support of others can anyone survive the ordeal. Fortunately, I have been blessed with an abundance of support from advisors, family, friends and colleagues. This thesis owes its existence to their kindness, patience, and encouragement.

I wish to thank my advisory committee, Dr. Michael Nelson, Dr. Yuri Quintana, and Dr. Robert Wood for their guidance and advice. I also wish to thank Dr. Bernd Frohmann for his assistance above-and-beyond the call of duty. I also thank the late Dr. Jean Tague-Sutcliffe for ensuring, despite her grave illness, that my thesis work would continue to thrive in her absence.

Drs. Shane Dunne and Mark Kinnucan are thanked for the never-ending generosity they have shown me in the sharing of their expertise. Kevin Kennedy is thanked for the care and attention he put into his programming tasks.

Finally, I must thank, a thousand times over, the members of my family. My mother, Marilyn Downie, is the rock upon which the foundations of this thesis, and all of my education, are laid. Elizabeth Downie, my aunt, has been especially kind in her ongoing financial and moral support. Finally, I thank my wife, Janet Eke, to whom I now dedicate this work. I also acknowledge that this dedication is small recompense for the unceasing love and patience she has extended to me over the course of this project.

Table of Contents

CERTIFICATE OF EXAMINATION ii
ABSTRACT iii
ACKNOWLEDGEMENTS iv
TABLE OF CONTENTS v
LIST OF TABLES ix
LIST OF FIGURES xi
LIST OF APPENDICES xii

1 INTRODUCTION 1

2 MUSIC INFORMATION 4

2.1 Introduction 4

2.2 Facets of music information 4
2.2.1 Pitch Facet 4
2.2.2 Temporal Facet 5
2.2.3 Harmonic Facet 6
2.2.4 Timbral Facet 7
2.2.5 Editorial Facet 8
2.2.6 Textual Facet 8
2.2.7 Bibliographic Facet 9

2.3 Representational completeness and MIR systems 9
2.3.1 Analytic/Production MIR systems 10
2.3.2 Locating MIR systems 10

2.4 McLane's "views" of musical "documents" 13

2.5 Implications for research 16

2.6 Summary 17

3 PREVIOUS RESEARCH 23

3.1 Introduction 23
3.1.1 Analytic/Production MIR systems 25
3.1.2 Locating MIR systems 29

3.2 Summary 37

4 THE MUSIFIND MUSIC INFORMATION RETRIEVAL PROJECT 39

4.1 Introduction 39

4.2 The MusiFind conception of a true music database 39

4.3 Operating paradigms of the MusiFind research project 40
4.3.1 Explication of Paradigm 1 40
4.3.2 Implications of Paradigm 1 41
4.3.3 Explication of Paradigm 2 43
4.3.4 Implications of Paradigm 2 44

4.4 Summaries of key MusiFind reports and publications 46
4.4.1 Creating the ideal full-text music database (Downie 1993b) 46
4.4.2 Implications of the MusiFind approach to music indexing 50
4.4.3 Name that tune: an introduction to music information retrieval (Tague-Sutcliffe et al. 1993) 52
4.4.4 Creating the ideal full-text music database: user assessment survey (Downie 1993d) 52
4.4.5 Colloquium: Toward the creation of a full-text music information retrieval system: a presentation of findings and future directions (Downie 1996b) 60
4.4.6 Concluding remarks on the MusiFind project 62

4.5 Summary 64

5 RESEARCH QUESTIONS AND STUDY OVERVIEW 65

5.1 Introduction 65

5.2 Principal hypothesis 66
5.2.1 Subsidiary hypothesis 1 67
5.2.2 Subsidiary hypothesis 2 67
5.2.3 Subsidiary hypothesis 3 67

5.3 Principal Components 67
5.3.1 Music database 67
5.3.2 Retrieval software 67
5.3.3 Phase I: Informetric analyses 68
5.3.4 Phase II: Information retrieval evaluation 69

5.4 Important changes 70
5.4.1 Selection of classification schemes 70
5.4.2 Selection of retrieval method 71
5.4.3 Selection of IR evaluation metrics (Dependent Variables) 72
5.4.4 Introduction of sensitivity evaluation (QQUAL) 74

5.5 Summary 75

6 PHASE I: INFORMETRIC ANALYSES 77
6.1 Introduction 77
6.2 Methods 78
6.2.1 Analytic tools 78
6.2.2 Descriptive statistics on intervals (type = interval) 78
6.2.3 Creation of classification schemes 79
6.2.4 Entropy of intervals in BD file 80
6.2.5 Entropy of intervals in n-grammed representations 81
6.2.6 Descriptive statistics on n-grams (type = n-gram) 83
6.2.7 Entropy of n-grams 84
6.2.8 Term discrimination analysis 85
6.2.9 Informetric modeling 86
6.3 Observations and analysis 87
6.3.1 Descriptive interval data and analyses 87
6.3.2 Descriptive interval data and analyses: concluding remarks 92
6.3.3 Descriptive n-gram data and analyses 93
6.3.4 Term discrimination data and analyses 112
6.3.5 Descriptive n-gram data and analyses: concluding remarks 116
6.3.6 Informetric modeling 118
6.4 Implications for Phase II retrieval evaluations 122
6.5 Summary 124
7 PHASE II: INFORMATION RETRIEVAL EVALUATIONS 125
7.1 Introduction 125
7.2 Methods 125
7.2.1 Analytic tools 125
7.2.2 Experimental design 126
7.2.3 Formal hypothesis concerning main effects 127
7.2.4 Sampling method 127
7.2.5 Query creation process 128
7.2.6 Error simulation procedure 130
7.2.7 Dependent measures 131
7.2.8 Analytic methods 131
7.3 Data and analyses 133
7.3.1 Descriptive data 133
7.3.2 Statistical analyses 139
7.3.3 Interpreting the multi-way interactions through QQUAL 143
7.4 Design recommendations 152
7.4.1 Informal questions regarding experimental factors: revisited 152
7.5 Salton's space-density *Q* and NPREC 154

7.6 Summary and conclusions 155
7.6.1 Principal hypothesis: revisited 156
7.6.2 Subsidiary hypothesis 1: revisited 157
7.6.3 Subsidiary hypothesis 2: revisited 158
7.6.4 Subsidiary hypothesis 3: revisited 158
7.6.5 Concluding remarks 158

8 CONCLUSION 160

8.1 Limitations of the study 162

8.2 Future research 163

APPENDIX A: REJECTED MODEL FITTING DATA 165

APPENDIX B: WITHIN-SUBJECT TESTS 167

BIBLIOGRAPHY 169

VITA 178

List of Tables

Table 4-1. Theoretical maxima of unique index terms 54

Table 4-2. Classification schemes 57

Table 4-3. Retrieval results (*incipit* queries) 59

Table 4-4. Retrieval results (random location queries) 62

Table 4-5. Retrieval results (contiguous string queries) 62

Table 5-1. Study nomenclature (Independent Variables) 66

Table 5-2. N-gram databases 68

Table 6-1. Descriptive data concerning intervals 79

Table 6-2. Summary of interval occurrence and classification schemes 79

Table 6-3. Frequency distortion factors caused by n-gramming 82

Table 6-4. Interval entropy values from n-grammed databases 90

Table 6-5. Descriptive data about n-gram tokens 94

Table 6-6. Number of n-gram types 94

Table 6-7. Descriptive data concerning n-gram types 95

Table 6-8. Music databases and standard test collections compared: Mean Types/Record 98

Table 6-9. Music databases and standard test collections compared: Mean (per record) Tokens/Type 99

Table 6-10. Entropy data concerning n-grams 102

Table 6-11. Number of songs in which the most frequently occurring n-gram type is found 107

Table 6-12. Top ranking types: text and n-grams compared 107

Table 6-13. Probability that a given n-gram occurs in *x* or fewer songs 112

Table 6-14. Term discrimination data 113

Table 6-15. GIGP (zero-truncated) model fitting data: View A 118

Table 6-16. GIGP (zero-truncated) model fitting data: View B 118

Table 6-17. C^2 values and best-fitting distributions 119

Table 7-1. Experimental factors 126

Table 7-2. Experimental design 127

Table 7-3. Error simulation rules 130

Table 7-4. NPREC descriptive summaries (averaged over queries of all lengths (QLEN) and both locations (QLOC)) 133

Table 7-5. NREC descriptive summaries (averaged over queries of all lengths (QLEN) and both locations (QLOC)) 133

Table 7-6. NPREC test of between-subjects effects (QLOC) 134

Table 7-7. NREC test of between-subjects effects (QLOC) 134

Table 7-8. NPREC: Select multivariate tests of within-subject effects 139

Table 7-9. NREC: Select multivariate tests of within-subject effects 140

Table 7-10. NPREC: Significant tests of within-subject contrasts 142

Table 7-11. NREC: Significant tests of within-subject contrasts 143

List of Figures

Figure 2-1. Facets of music information, Example 1 18

Figure 2-2. Facets of music information, Example 2 19

Figure 2-3. *Incipit* index. 20

Figure 2-4. Thematic *incipit* index 21

Figure 2-5. Notation index.. 22

Figure 3-1. *RISM* database indexes 30

Figure 3-2. Encoding examples, *RISM* database.. 31

Figure 3-3. *MELDEX* retrieval results 33

Figure 6-1. Distribution of interval tokens over interval types 78

Figure 6-2. N-grams of intervallic information 82

Figure 6-3. Rank-probability comparison of intervals and letters. 90

Figure 6-4. Distribution heads compared: text and music n-grams (View A) 110

Figure 6-5. Percentage of positive and negative discriminators 114

Figure 6-6. Document space density *Q* values for each database 115

Figure 6-7. Three sample records (abridged) from the CACM test collection 120

Figure 7-1. NPREC by QLOC 135

Figure 7-2. NPREC by CLASS 135

Figure 7-3. NPREC by NLEN 136

Figure 7-4. NPREC by QLEN 136

Figure 7-5. NREC by QLOC 137

Figure 7-6. NREC by CLASS 137

Figure 7-7. NREC by NLEN 138

Figure 7-8. NREC by QLEN 138

List of Appendices

APPENDIX A REJECTED MODEL FITTING DATA 165

Table A-1. Rejected View A model fitting data..... 165

Table A-2. Rejected View B model fitting data..... 166

APPENDIX B WITHIN-SUBJECT TESTS 167

Table B-1. NPREC: Multivariate tests of within-subject effects..... 167

Table B-2. NREC: Multivariate tests of within-subject effects..... 168

1 Introduction

Before the advent of digital computers, the principal method of accessing music information was the *thematic* catalogue. Scholars and musicians have consulted these printed volumes for over a thousand years (Brook 1980). In them, they have found fragments of musical works called *incipits* that represent the beginnings of a work, or significant parts (i.e, themes). These *incipits* have taken on various forms including "conventional notes, neumes, tablatures, numbers, letters or computer codes" (Brook 1980). The amount of information conveyed by *incipits* can vary depending on their representation (see Chapter 2). Sometimes *incipits* have been verbatim extracts from a musical score and contain pitch, harmonic, rhythmic, editorial, textual, and timbral information. Other times, the authors of thematic catalogues have seen fit to greatly reduce the amount of information presented by representing only select aspects of a melody, usually pitch names (e.g., Barlow and Morgenstern 1949). Information has been further reduced by representing *incipits* not through the use of the pitches, but by intervals (i.e., the distance between notes) (e.g., Parsons 1975). Despite the method of representation used, the one thing that makes thematic catalogues special is that they attempt to give users the ability to access music information on its own terms; that is, ***the ability to answer music queries framed musically***.

Automating access to music information through the use of digital computers has intrigued musicologists, computer scientists, librarians and music lovers alike. Each has his own purpose in mind and thus there seems to be as many approaches to developing Music Information Retrieval (MIR) systems as there are users. A half-hour's perusal of the back issues of *Computing in Musicology* (Hewlett and Selfridge-Field, eds.) will bring this fact to the fore. Some have designed complex suites of computer tools (e.g., Huron 1991) to analyze all the varied facets of music. Others have tried to automate the thematic catalogue by including *incipits* as part of a bibliographic record (e.g., *RISM* 1997).[1] Still others have explored the idea of using sophisticated approximate string

[1] *Répetoire International des Sources Musicales.* See Chapter 3.1.2 for more information.

matching techniques (e.g., Ghias et al. 1995; McNab et al. 1996, 1997; Prechelt and Typke 1998). One thing that unites all of these approaches is that they have some kind of shortcoming. The more powerful analytic systems can be very difficult to use, *incipit* indexes leave out large amounts of music that might be of interest, and approximate string-matching techniques can be computationally expensive without necessarily giving better results (see Chapter 3).

Taking our cue from those printed thematic catalogues that have reduced the amount of music information represented (e.g., Parsons 1975; Keller and Rabson 1980) we developed, and then evaluated, a MIR system based upon the intervals found within the melodies of a collection of 9354 folksongs (see Chapters 4-8). We believe that there is enough information contained within an interval-only representation of monophonic melodies that effective retrieval of music information has been achieved. We extended the thematic catalogue model by affording access to musical expressions found anywhere within a melody. To achieve this extension we fragmented the melodies into length-n subsections called n-grams (see Chapter 4.4.1.1). The length of these n-grams and the degree to which we precisely represent the intervals are variables analyzed in this thesis.

N-grams form discrete units of melodic information much in the same manner as words are discrete units of language. Thus, we have come to consider them "musical words." This implies that, for the purposes of music information retrieval, we can treat them as "real words" and thereby apply traditional text-based information retrieval techniques. We examined the validity of our "musical word" concept in two ways. First, a variety of informetric analyses were conducted to examine in which ways the information properties of "musical words" and "real words" are similar or different (Chapter 6). Second, we constructed a collection of "musical word" databases using the famous text-based, SMART information retrieval system (see Salton and McGill 1983; Salton 1989). A group of simulated queries was run against these databases. The results were evaluated using the normalized precision and normalized recall measures (see Chapter 7).

This study is significant and adds to knowledge in two ways (see Chapter 8). First, our concept of “musical words” shows great merit thus implying that useful MIR systems can be constructed simply and efficiently using pre-existing text-based information retrieval software. Second, this study represents the first formal and comprehensive evaluation of a MIR system of any type using rigorous statistical analyses to determine retrieval effectiveness (see Chapter 3).

2 Music Information

2.1 Introduction

In this chapter, we will examine the seven facets of music information that might be of interest in the context of Music Information Retrieval (MIR). We will frame the development of MIR systems as one of representational completeness and not the choice of a particular Music Representation Language (MRL). We will categorize MIR systems into two classes: *Analytic/Production* and *Locating* MIR systems. As *Locating* MIR systems are the object of this research, we will enumerate their features in some detail. We will show that *Locating* MIR systems do not require, nor necessarily benefit from, representational completeness. Finally, we will justify the limited focus of this study based upon the facts brought forth in this chapter.

2.2 Facets of music information

The author has identified seven facets of music information that play various roles in defining the domain of Music Information Retrieval. These are the Pitch, Temporal, Harmonic, Timbral, Editorial, Textual and Bibliographic facets. Due to complexities inherent in the representation of music information, this is not a facet analysis in the strict sense because the facets are not mutually exclusive. For example, the term *adagio* when found in a score could be placed within both the Temporal and Editorial facets, depending on context. This explication and analysis is meant to highlight some of the significant information aspects that might have value for the user of a MIR system, along with the problems posed by the nature of music information in the storage and retrieval of that information.

2.2.1 Pitch Facet

Pitch is “the perceived quality of a sound that is chiefly a function of its fundamental frequency in --the number of oscillations per second ” (Randel 1986). The graphical representation (e.g.,  , , , , etc.) where pitch is represented by the vertical position of a note on the staff is the most familiar. Note names (e.g., A, B, C#, ..., etc.), scale degrees (e.g., I, II, ...,VII), solfège (e.g., do, ré, ..., ti) and pitch-class numbers (e.g., 0, 1, 2, 3, ..., 11) are also some of the many methods of representing pitch.

The difference between two pitches is called an interval. Intervals can be represented by the signed difference between two notes as measured in semitones (e.g., -8, -7…, -1, 0, +1, ..., +3, etc.) or by its tonal quality as determined by the location of the two pitches within the syntax of the Western theoretical tradition. For example, the interval between A and C# is called a Major 3rd while the aurally equivalent distance between A and D♭ is a Diminished 4th. Melodies can be considered sets of either pitches or intervals perceived as being sequentially ordered through time.

The notion of key is included here as a sub-facet of pitch. The melodic fragment EDCEDC (i.e.,"Three Blind Mice") in the key of C Major is considered to be musically equivalent to BAGBAG in the key of G Major. That is to say, that the melodic contours (i.e., the pattern of intervals) are perceived by the listener to be equivalent, despite the fact that the absolute pitches of the latter are higher than the former. In our experience, singers are the most sensitive to the notion of key, for they must find works, or transpositions of works, in a key that does not extend the absolute pitches of a melody beyond their particular vocal ranges.

2.2.2 Temporal Facet

Information concerning the duration of musical events falls under the Temporal facet. This includes tempo indicators, meter and pitch durations. Taken together these three make up the rhythmic component of a musical work. Rests in their various forms can be considered indicators of the duration of musical events that contain no pitch information. Temporal information poses significant representational and access problems. Temporal information can be absolute (e.g., a metronome indication of MM=80), general (e.g., *adagio, presto*, *fermata*, etc.) or relative (e.g., *schneller, langsamer*, etc.). Temporal distortions are sometimes encountered (e.g., *rubato, accelerando*, *rallentando*, etc.). Because the rhythmic aspects of a work are determined by the complex interaction of tempo, meter, pitch duration, as well as, accent (whether denoted or implied), it is possible to represent a given rhythmic pattern many different ways, all of which yield aurally identical results. Some performance practices, where it is expected that the player(s) will deviate from the strict rhythmic values noted in the score (e.g., Swing, Jazz, etc.), give rise to added complexities, similar to those caused by the

temporal distortions mentioned above. Thus, representing temporal information for retrieval purposes is particularly problematic.

2.2.3 Harmonic Facet

When two or more pitches occur at the same time, a simultaneity, or harmony, is said to have occurred. This is also known as polyphony, while absence of polyphony is called monophony. Pitches that align vertically in a standard Western score are creating harmony. The interaction of the Pitch facet and the Temporal facet to create polyphony is a central feature of Western music. Over the centuries music theorists have codified the most common simultaneities (i.e., pitches that sound at the same point in time) into several comprehensive representational systems based upon their constituent intervals and the perceived function of those intervals within the syntax of the modality of work in which they occur. Theorists have also codified the common sequential patterns of simultaneities found within Western tonal music. While it is beyond the scope of this present work to examine in detail the complex realm of Western harmonic theory and praxis, suffice it to say that an individual harmonic event can be denoted by a combination of the interval(s) it contains and the scale position of its "root," or fundamental, pitch. A chord, like that sounded when a guitar is strummed, is an example of an harmonic event. Sequences of chords, or harmonic events, can be represented by chord names. The very common harmonic sequence, or progression, in the key of C Major, [C+ F+ G+ C+] is here represented by the note name of the fundamental pitch of each chord. The "+" denotes that each chord contains the intervals of Major 3rd and Perfect 5th as measured from the fundamental note. Another method of representing this harmonic progression, that generalizes it to all major keys, is to indicate the scale degree of the root of the chord using Roman numeral notation: **I-IV-V-I**.

Simple access to the codified aspects of a work's harmonic information can be problematic because its harmonic events, while present in the score, are not usually denoted explicitly in one of the ways described above.[2] The matter is further complicated

---

[2] Exceptions to this are the inclusion of chord names or chord symbols in most popular sheet music, and the harmonic shorthand, called *basso continuo*, commonly found in music from the Baroque period (see Figure 2-1, *Cembalo* part).

by the fact that the human mind can perceive and consistently name one of the codified simultaneities, despite the presence of extra pitches called non-chord tones. Even with the absence, or delay, of one or more of the chord's constituent pitches, most members of Western societies can still consistently classify the chord.

2.2.4 Timbral Facet

The Timbral facet comprises all aspects of tone-colour. The aural distinction between a work played upon a flute and upon a violin is caused by the differences in timbre. Thus, orchestration information, that is, the designation of specific instruments to perform all, or part, of a work, falls under this facet.[3] A wide range of performance methods also affects the timbre of music (e.g., *pizzicatti*, mutings, pedalings, bowings, etc.). Here the border between timbral and editorial information becomes blurred as these performance methods can also be placed within the editorial facet. The act of designating a performance method that affects timbre is editorial; the aural effect of the performance of the chosen method is timbral. Because timbral information is best conveyed in the aural representation of the work, accessing timbral information through a timbral query (i.e., playing a muted trumpet and asking for matches), would require signal processing capabilities that are beyond the scope of this thesis.[4] A more practicable, yet still difficult solution, would be to access timbral information through some type of interpretation of the editorial markings. This possible solution would, of course, be subject to the same difficulties associated with representing editorial information, discussed below.

---

[3]In practice, orchestration information, while really part of the Timbral facet, is sometimes considered part of the Bibliographic facet. The simple enumeration of the instruments used in a composition is usually included as part of a standard bibliographic record. This information has been found to assist in the description, and thus the identification, of musical works.

[4] Progress is being made in this area. Muscle Fish, an audio engineering firm, now offers a database plugin, called *Audio Information Retrieval* (AIR) *Datablade* which is capable of timbral matching within a limited domain (Muscle Fish, http://www.musclefish.com).

2.2.5 Editorial Facet

Performance instructions make up the majority of the Editorial facet. These include fingerings, ornamentation, dynamic instructions (e.g., ***ppp***, ***p,*** ...***f***, ***fff***), slurs, articulations, *stacatti*, bowings, and so on. The vagaries of the Editorial facet pose numerous difficulties. One difficulty associated with editorial information is that it can be either iconic (e.g., **-** , **3**, **!** , etc.), or textual (e.g., *crescendo*, *diminuendo*, etc.), or both. Furthermore, editorial information can also include the parts of the music itself. The writing out of the harmonies from the *basso continuo*, also known as the "realization of the figured-bass," is an editorial act. (See Figure 2-1, *Cembalo* part.) *Cadenzi* and other solos, originally intended by many composers to be improvised, are frequently realized by the editor. Lack of editorial information is yet another problem to be considered. Like the *basso continuo*, where the harmonies are implied, many composers have simply assumed that the performers were competent to render the work in the proper manner without aid of editorial information. In many cases, the editorial discrepancies between editions of the same work make the choice of a "definitive" version of a work for inclusion in a MIR system very problematic.

2.2.6 Textual Facet

The lyrics of songs, *arias*, chorales, hymns and symphonies, etc., are included in the Textual facet. *Libretti*, the text of operas, are also included. It is important to note that the Textual facet of music information is more independent of the melodies and arrangements that are associated with it than one would generally believe. A given lyric fragment is sometimes not informative enough to identify and retrieve a desired melody and *vice versa* (Temperley 1993). There is a strong tradition in Western music of freely interchanging lyrics and music. A perfect example of this phenomenon is the tune, "God Save the Queen." Known to citizens of the British Commonwealth as their royal anthem, this simple tune is also known to Americans as their republican song, "America." In Figure 2-2, note that the lyrics are not those present-day Canadians would associate with their national anthem, "O Canada." Also note the sequence of numbers located near the title: **10 10 8 6 8 6 8 10**. These numbers tell the user that any other set of lyrics might be substituted for those given, provided that the number of syllables per line matches the

given sequence. Many songs have also undergone translation into many different languages. Simply put, one must be aware that a given melody might have multiple texts and that a given text might have multiple musical settings. It is also important to remember that there exists an enormous *corpus* of music without any text whatsoever.

2.2.7 Bibliographic Facet

Information concerning a work's title, composer, arranger, editor, lyric author, publisher, edition, catalogue number, publication date, discography, performer(s), and so on, are all aspects of the Bibliographic facet. This is the only facet of music information that is not derived ***from*** the content of a composition; it is, rather, information, in the descriptive sense, ***about*** a musical work. It is music metadata. All of the difficulties associated with traditional bibliographic description and access also apply here. Howard and Schlichte (1988) outline these problems along with some of their proposed solutions. Temperley (1993) is another important work tackling this difficult subject.

2.3 Representational completeness and MIR systems

Alexander McLane's chapter in the 1996 *Annual Review of Information Science and Technology* entitled *Music as Information* is a superlative review of the many Music Representation Languages (MRLs) that have been developed, or proposed, for use in MIR systems (McLane 1996). For those interested in a technical comparison of the attributes of five of the most important MRLs, we recommend Selfridge-Field (1994).[5] It is not the purpose of this thesis to evaluate the relative merits of individual MRLs. What is of interest, however, is the degree of "representational completeness" required to create various MIR systems. We define the degree of "representational completeness" by the number of music information facets (and their sub-facets) included in the representation of a musical work, or group of works. A representation that includes all the music information facets (and their sub-facets), in some form, is "representationally complete."

In general, MIR systems can be grouped into one of two categories: *Analytic/Production* MIR systems and *Locating* MIR systems. The latter is the object of

---

[5]This article presents, in easy-to-understand tabular form, how the facets of music information are (or are not) represented in the MuseData, DARMS, SCORE, MIDI and Kern MRLs.

this thesis. The two types of MIR systems can be distinguished by 1) their intended uses; and, 2) the representational completeness of music information found within. Of the two, *Analytic/Production* systems require by far the most complete representation of music information. Working descriptions of the two types of MIR systems are given below. More detailed examples of each are presented in the next chapter.

2.3.1 Analytic/Production MIR systems

Intended users of *Analytic/Production* MIR systems include such experts as musicologists, typesetters, composers, etc. These are MIR systems that have been designed with the goal of being as representationally complete as possible. For the most part, designers of such systems wish to afford fine-grained access to all the aforementioned facets of music information, with the possible downplaying of the Bibliographic facet. Fine-grained access to music information is required by musicologists to perform detailed theoretical analysis of, for example, the melodic, harmonic, or rhythmic structures of a given work, or body of works. Typesetters need fine-grained access to assist them in the efficient production of publication-quality musical scores and parts. Composers make use of fine-grained access to manipulate the myriad musical elements that make up a composition. Because of the storage and computational requirements associated with high degrees of representational completeness, *Analytic/Production* systems usually contain far fewer musical works than *Locating* MIR systems.

2.3.2 Locating MIR systems

Locating MIR systems are those systems that have been designed to assist in the identification, location, and retrieval of musical works. Text-based analogues include Online Public Access Catalogues (OPACs), and full-text, bibliographic information retrieval (FBIR) systems, like those provided by the DIALOG collection of databases.[6] Intended users are expected to have a wide range of musical expertise, ranging from musically naive persons to such experts as musicologists and other musical professionals. For the most part, users wish to make use of the musical works retrieved, for either

---

[6] FBIR systems are those which combine both full-text content (or extensive abstract surrogates) and structured bibliographic metadata

performance or audition, rather than analyzing or manipulating the various facets of the music information contained within the system.[7] Thus, the objects of retrieval can be considered to be more coarsely-grained than those associated with *Analytic/Production* MIR systems. Because the objects of retrieval are more coarsely-grained, access points to the music information are traditionally limited to various combinations of select aspects of the Pitch, Temporal, Textual and Bibliographic facets. The following explication will help clarify the characteristics of a *Locating* MIR system.

2.3.2.1 Uses of a *Locating* MIR system[8]

Some queries in the field of music are text-based and parallel those in other fields. The Bibliographic and Textual facets of music information can be used to answer the following queries:

1) List all compositions, or all compositions of a certain form, by a specified composer
2) List all recordings of a specified composition, or composer
3) List all recordings of a specified performer
4) Identify a song title given the first line of lyrics, or vice versa.

A good review of the role the computer has played in improving retrieval from textual catalogues of musical scores and discographies, is found in Duggan (1992). She points out, for example, that OCLC [9] contains catalogue records for 606,000 scores and 719,000 sound recordings, and the Music Library CD-ROM published by Silver Platter contains more than 408,000 records for sound recordings. However, the ability to store

---

[7]While some users might wish to perform some kind of analysis of the data contained within a *Locating* MIR, this is not customarily a design consideration. Many information scientists have performed informetric analyses on the information contained within OPACs and FBIRs (e.g.,Wolfram 1992a,b; Nelson 1988); however, designers of such text-based systems have not felt compelled to take such analytic uses into consideration when creating their systems.

[8] Adapted from Tague-Sutcliffe et al. (1993).

[9] Online Computer Library Center, the major source of computerized catalogue records for most libraries.

some searchable representation of the music itself provides the user with the capability of answering queries beyond those served by a Machine Readable Cataloguing (MARC) format bibliographic catalogue:

5) Given a composer, identify by the first few bars each of his or her compositions, or compositions of a certain type

6) Given a melody, for example the tune of a song or the theme of a symphony, identify the composition or work.

The first of these two types of queries has traditionally been answered by means of printed *incipit* indexes, typically simple listings of the beginning bars of the scores in a particular collection. Figure 2-3 is a good example of a printed *incipit* index (Edson 1970).

The second type of query has traditionally been answered by thematic indexes to musical compositions. An example of such an index (Barlow and Morgenstern 1949) is shown in Figure 2-4. The book contains a few bars of one or more themes from 10,000 musical compositions, arranged by composer. A "Notation Index" in the back of the book permits the user to look up a sequence of six to eight notes, transposed into the key of C, as an alphabetical listing of transposed 'themes' to identify the composition in which it occurs (Figure 2-5).

Some *Locating* MIR systems are best considered automated *replications* of *incipit* and thematic indexes: the *RISM* (1997) database and Prechelt and Typke's *Tuneserver* (1998), both examined in the next chapter, are good examples. Other systems, like that discussed in McNab et al. (1997) and the one created by us, can be considered automated *extensions*. They are extensions in the sense that they have not limited themselves to the storage and access of *incipits* and themes; rather, they attempt to exploit the information found in some machine-readable "full-text" representation of the music.[10] The greatest advantage to extending the traditional *incipit* and thematic

---

[10]Here "full-text" is used in the sense that melodic information is not arbitrarily truncated (as it is in *incipit* and thematic indexes). For example, Parsons' (1975) index contains no melodic string longer than 15 notes.

indexes to include “full-text” information is that memorable music events can occur anywhere within a work and many potential queries will reflect this fact (McNab et al. 1996). Thus, when “full-text” access is made possible, a *Locating* MIR system could also satisfy the following queries:

7) In which compositions can we find the following note sequence anywhere in the composition?

8) Which composers have used the following combination of instruments in the orchestration of a passage?

To summarize, the two types of MIR systems differ significantly both in their intended uses, and in their levels of representational completeness. Furthermore, if one considers a high degree of representational completeness to be *depth,* and the number of musical works included to be *breadth,* then *Analytic/Production* MIR systems tend toward depth at the expense of breadth, while *Locating* MIR systems tend toward breadth at the expense of depth.

2.4 McLane’s “views” of musical “documents”

McLane’s (1996) classification of musical representations is very useful in the present context. He classifies representations of musical works, or “documents,” into three “views”: the *subjective*, the *objective,* and the *interpretative*. Below is our summation and adaptation of McLane’s categories:

> ***Subjective*** **view**. The use of a notation scheme to represent a musical work can be considered a subjective view of that work. It is subjective because the choice of notational elements used to represent a work is context-dependent in the sense that it is the notator’s decision to include or exclude particular aspects of the work. Representational completeness can range from the minimalist representation of musical extracts like those found in the aforementioned "Notation Index," to the full orchestral score of Beethoven’s Ninth Symphony. While McLane does not, we include descriptive bibliographic information as part of the subjective view.

***Objective* view**. A sound recording yields an objective view of a musical work. It is objective because once recorded, the music represented by the recording is fixed and no longer subject to editorial or performance variations. This view can be considered the most complete representation, as it includes all the information from the Pitch, Temporal, Harmonic, Editorial, and Timbral facets.

***Interpretative* view**. The representation of a work through the analysis of some aspect(s) of the work is the interpretative view of that work. Classificatory and analytic schemes that elucidate otherwise non-obvious features of a musical work, or group of works, fall within this category. Critical evaluations, such as those found in music reviews, are also part of the interpretative view.

According to McLane (1996):

> Any representation of music will consist of one or more of these three views, all of which will face similar questions concerning *how much* of the original "document" is *necessary for the purposes of retrieving information* useful to musical analysis. [Italics mine]

The above statement also applies equally to the retrieval of music information for purposes of identification, location, and retrieval.

Of McLane's three views, the *objective* would best suit the needs of those wishing to develop the ultimate MIR system of either type, for it is representationally the most complete.[11] Unfortunately, signal processing systems capable of parsing a sound recording of a work into all of its constituent music information facets do not yet exist.

The *interpretative* view is of limited interest in the present discussion, for it is concerned with musical meta-information. As meta-information, this view "bypasses much of what is contained in the other views," [and] "...lacks their flexibility," [which]"…requires greater knowledge on the part of the user" (McLane 1996).

Having eliminated the other two views from further consideration, we will focus the remainder of this discussion on the implications of adopting the subjective view and

---

[11]Actually, with the inclusion of bibliographic information, the ultimate MIR system would provide both *Analytic/Production* and *Locating* functions

its varying degrees of representational completeness. Consider now, just how incomplete a representation of a given work is provided by the "Notation Index": it contains only a simplistic, subjective representation of the Pitch facet. Missing from this representation is all key, harmonic, temporal, editorial, textual, timbral and bibliographic information. The *National Tune Index* (Keller and Rabson 1980) offers two similarly minimalist representations of musical *incipits*: scale degree (represented by number) and interval-only sequence (represented by signed integers). Lincoln's (1989) index of Italian madrigals also contains an interval-only (signed integers) representation of the *incipits* it contains. The index developed by Parsons (1975) reduces the degree of representational completeness to an extreme. His index represents musical *incipits* as strings of intervals using text strings containing only four symbols: *, **R**, **U**, and **D**.[12]

Obviously, such incomplete representations would have very limited use in an *Analytic/Production* MIR system. However, as locating tools these minimalistic representations have shown themselves to have great merit. In fact, it is the incompleteness of their music representations that makes them effective as access tools. By minimizing the amount of information contained in the representation of the *incipits*, these indexes also reduce the need for the user to come up with more representationally complete queries. Thus, the musically naive user can make use of these representations with mimimal opportunities for introducing query errors. Furthermore, should an error be introduced, it is less likely to result in an identification or retrieval failure (see Chapter 4 for an explanation). Thus, for the purposes of identification, location, and retrieval, that is to say, the creation of *Locating* MIR systems, it is not necessary, nor necessarily desirable, to have representational completeness.

This conclusion is supported by McLane (1996), who is writing about *Locating* MIR systems when he states:

> Both the choice of view from a representation of music and the degree of completeness of a work's representation depend on the user's information needs. Information retrieval is an interactive process that depends on the

---

[12]Parsons' symbols: ***** indicates *incipit* beginning; **R** for note Repeats (interval of 0 semitones); **U** for Up (any positive interval); and **D** for Down (any negative interval).

> knowledge of the user and the level of complexity of the desired information. In the case of the need for the simple identification of a musical work where bibliographic information is not unique enough, one may limit the view to a subjective one involving a relatively small subset of the notated elements of the work, often the pitches of an opening melodic phrase. The representation of pitches will be in a form that the user is likely to expect and be able to formulate a query using the same terminology, or at least one that is translatable into the form of the representation.

We have concluded that representational completeness is not a prerequisite for the creation of a useful *Locating* MIR system. However, why is it that music information tends to be reduced to simplistic representations of the Pitch facet for retrieval purposes? Why not use simple representations of the Rhythm facet? or perhaps, the Timbral facet? McLane's (1996) comments above, and the decisions by Barlow and Morgenstern (1949), Parsons (1975), Keller and Rabson (1980), and Lincoln (1989) to adopt a minimal degree of representation in their music indexes (i.e., simplistic representations of the Pitch facet), were not arbitrary. Pyschoacoustic research has shown the contour, or shape, of a melody to be its most memorable feature (Kruhmhansl and Bharucha 1986; Dowling 1978). Thus, any representation that highlights a work's melodic contour (i.e., sequences of intervals) while filtering out extraneous information (i.e., exact pitches, rhythmic patterns, etc.) should, in theory, increase the chances for the successful identification, location and retrieval of a musical work.[13]

2.5 Implications for research

The author has chosen to examine the retrieval characteristics of various interval-only representations of melodic information. If one imagines the various indexes discussed above as a lying on a representational-completeness continuum, from Parsons' extremely miminalist representation to Barlow and Morgenstern's more moderate reduction in representational completeness, then our representations can be placed in the middle of the continuum, alongside Lincoln's, and Keller and Rabson's interval-only indexes. Thus situated, both past practice and psychoacoustic research support the hypothesis that some, or all, of our interval-only representations have the *potential* for

---

[13] Parsons' index captures this feature the best.

effective music information retrieval. Whether this hypothesis is true, is of course, subject to testing. Hence, this thesis.

2.6 Summary

In this chapter, we have examined the seven facets of music information that might be of interest in the context of MIR. Of these, six facets (Pitch, Temporal, Harmonic, Timbral, Editorial, and Textual) denote information found *within* musical works. The seventh, Bibliographic facet, is information *about* musical works. We have highlighted some of the problems associated with reliance on the various facets as retrieval points. We have framed the development of MIR systems as one of representational completeness and not the choice of a particular MRL. We have defined the degree of representational completeness as the number of music information facets (and their sub-facets) included in a MIR system. We have categorized MIR systems into two classes: *Analytic/Production* and *Locating* MIR systems. Each type of system has its own intended uses and requires different levels of representational completeness. We have shown that *Locating* MIR systems do not require, nor necessarily benefit from, representational completeness. We have justified limiting the focus of our research to an evaluation of the information retrieval characteristics of interval-only representations of the Pitch facet both in terms of past practice and psychoacoustic research.

Figure 2-1. Facets of music information, Example 1. Antonio Vivaldi's Concerto grosso in G minor, Opus 3, No. 2. Source: Palisca (1980).

Figure 2-2. Facets of music information, Example 2. *O Canada.* Source: United Church of Canada (1930).

Figure 2-3. *Incipit* index. Source: Edson (1970).

Figure 2-4. Thematic *incipit* index. Source: Barlow and Morgenstern (1949).

Figure 2-5. Notation index. Source: Barlow and Morgenstern (1949).

3 Previous Research

3.1 Introduction

The literature of Music Information Retrieval research is primarily one of development, not evaluation. That which is not present speaks louder than what is. We know of only one formal information retrieval (IR) evaluation of a MIR system. By "formal IR evaluation" it is meant those studies of the kind usually performed within the discipline of information retrieval as described by Tague-Sutcliffe (1992), Keen (1992), Korfhage (1997), and most definitively by Harter and Hert (1997).

This is not to say that that the MIR literature is completely devoid of its evaluative components. However, with the exception of Uitenbogerd and Zobel (1999) and their yet-to-be published technical report, there are no studies of system performance based upon the traditional "Cranfield Model" of IR evaluation. The "Cranfield Model" is named after the series of experimental IR evaluations performed by Cleverdon et al. (1966) in Cranfield, England, during the 1960s. The second of the Cranfield studies (Cranfield II) has been called "the exemplar for experimental evaluation of information retrieval systems" (Harter and Hert 1997). Thus, the Cranfield experiments are considered by most IR researchers to be the progenitor of the discipline of IR evaluation. The Cranfield Model of evaluation has the following principal components:[14]

1) A test collection of documents

2) A set of queries

3) A set of relevance judgements

For a given query and act of retrieval, then the following document sets and associated numbers are known:

A) Relevant and retrieved documents

B) Nonrelevant and retrieved documents

---

[14] This extended description of the Cranfield Model is taken, slightly modified, from Harter and Hert (1997).

C) Relevant and nonretrieved documents

D) Nonrelevant and nonretrieved documents

> Thus the performance of the IR system (in Cranfield, systems were viewed as indexing systems) for a given act of retrieval (query, document collection, retrieved documents, and associated relevance judgments) can be precisely assessed by examining the output produced by the system and computing measures based on this output. In the Cranfield tests the principal measures of performance were:
>
> **RECALL** = $a/(a + c)$; and,
>
> **PRECISION** = $a/(a + b)$

In addition to Uitenbogerd and Zobel (1999), only three papers, McNab et al. (1996, 1997) and Prechelt and Typke (1998), all discussed later in this chapter, contain reports of system evaluations of the sort similar to that defined above. References to other evaluation studies are also absent from the aforementioned trio of papers.[15] It is important to note that only Uitenbogerd and Zobel (1999), however, used the evaluative measures of recall and precision. McLane's (1996) review of the literature also confirms the paucity of formal evaluation research, the final paragraph of which is most telling in the present context. After extended email communications with this author during the Autumn of 1995, McLane concluded that the research endeavours outlined in this thesis (including formal "Cranfield Model" evaluations of performance) had not yet been undertaken. The concluding paragraph of his literature review is reproduced in its entirety below:

> What has been left out of this discussion, and will no doubt be a topic for future study, is the potential for applying some of the standard principles of text information retrieval to music representations. These are enumerated by SALTON & MCGILL and by LOSEE and include Boolean operations*, inverted file systems*, text analysis*, theories of document similarity*, various types of retrieval and their evaluation* and refinement, natural-language processing, and clustering. It remains to be seen, for instance, whether such aspects of text analysis as the frequency of use of words*, their average distance from each other*, and the

[15] We find it heartening that Uitenbogerd and Zobel (1999) make reference to Downie (1995).

> relationship of frequency to rank*, will apply equally well to musical analysis, where the concept of a lexical unit corresponding to a Aword@ may have more meaning in some musical works than others*. The application of these principles to music will in time be integrated with other nontext formats, particularly those that are time based, such as video, resulting in the broadening in scope of many of the traditional concepts of information retrieval [Asterisks mine].

Each item denoted by an asterisk has undergone evaluation within the context of this thesis.[16] Thus, what McLane appears to be suggesting is that the research project outlined in this thesis represents an examination of uncharted intellectual territory.

In this chapter, we will highlight some of the more significant approaches taken to develop MIR systems. Certain commonalties will come to the fore. These include:

1) the predominance of regular-expression matching procedures that use linear scans of the music databases;

2) the near total absence of traditional indexing methods to enhance retrieval performance;

3) the compilation of many small, task-specific programmes into larger MIR *toolkits*; and,

4) in the case of Locating MIR systems, the predominance of *incipits* as the access medium.

The extensive MIR work of the author and his associates is not reviewed in this chapter, but in Chapter 4. Prior research projects that pertain to questions of research method can be found in Chapters 5, 6, and 7.

3.1.1 Analytic/Production MIR systems

Three of the items summarized here are Doctoral dissertations: Rubenstein (1987), McLean (1988) and Page (1988). This fact is noted, for as dissertations they should, at the time of their respective defences, represent the state-of-the-art in MIR development better than any other publications.

---

[16]Details about each are given in Chapters 6 and 7.

Rubenstein (1987) extended the classic entity-relation model to include two novel features: *hierarchical ordering* and *attribute inheritance*. These novel features allowed Rubenstein to propose the creation of representationally complete databases of music using the relational database model. Each element of music information would be broken down into ever more fundamental units and these units would then participate in complex relations with each other. The extraordinary number of entities required to realize his model meant that an operational system was never implemented. Rubenstien=s proposal to exploit the performance enhancing characteristics of A-tree indexes to speed up searching is worth noting: it is one of the few instances in the literature where the use of indexes instead of linear scanning is explicitly suggested.

McLean (1988) is another researcher who has argued for the need to improve retrieval performance. His doctoral project was "...the definition of a complete representation of musical scores at the level of internal representation"(McLean 1988). He concluded that the "...traversability [of the musical data] via multiple sequential and direct access indexing schemes …" is a part of the "...necessary set of database-level services…" required for the creation of *Analytic/Production* MIR systems (McLean 1988). Other than a brief discussion of the usefulness of doubly-linked lists, he does not make it clear how he would implement such indexing schemes.

Page (1988) implemented an experimental system with "a restricted query system capable of retrieving melodic and rhythmic patterns." While he also mentions that some type of indexing would improve retrieval performance, his system uses a query language based on regular expressions. The musical data is searched through using specially designed Finite State Automata. Items of interest are retrieved via a single-pass, linear traversal of the database.

One goal of Page=s doctoral thesis was to map out the necessary components of a Amusical researcher=s toolkit" (Page 1988). Many of today=s *Analytic/Production* systems are best thought of as suites of computer tools. Each tool is designed to address one of the many processes involved in the creation, and use, of a MIR system. Tools include encoding programmes, extraction programmes, pattern-matching programmes, display programmes, data conversion programmes, analysis programmes, and so on.

David Huron's *Humdrum Toolkit* is an exemplar of its type. It is a collection of over 50 interrelated programmes designed to take advantage of the many information processing capabilities found in the UNIX operating system. Taken together, these tools create an incredibly powerful MIR system where "queries of arbitrary complexity can be constructed" (Huron 1991). Interest in his system is high and courses on its use are regularly offered. Huron (1991) best describes *Humdrum*'s flexibilities:

> The generality of the tools may be illustrated through the *Humdrum pattern* command. The pattern command supports full UNIX regular-expression syntax. Pattern searches can involve pitch, diatonic/chromatic interval, duration, meter, metrical placement, rhythmic feet, articulation, sonorities/chords, dynamic markings, lyrics, or any combination of the proceeding as well as other user-defined symbols. Moreover, patterns may be horizontal, vertical, or diagonal (*i.e.*, threaded across voices).

Like most things in life, all of this power comes at a price. Korsnstädt (1996) provides a brilliant example of how *Humdrum*'s Unix-style command-line interface

> ...minimizes the number of potential users. For example, in order to search for occurrences of a given motive and to annotate the score with corresponding tags, the user has to construct the following command:
>
> ```
> extract –i'**kern' HG.kern | semits -x | xdelta -
> s = | patt -t Motive1 -s = -f Motiv1.pat |
> extract –-i**'patt' | assemble HG.krn
> ```
>
> The construction of such a command requires a substantial facility in the use of UNIX tools.

It is hard to imagine the naïve user ever managing to formulate such a query. Although work is underway to make *Humdrum* more user-friendly (Kornstädt 1996), it must be stressed that *Humdrum* is intended for use by musically sophisticated users who need analytic power more than they need syntactic simplicity. Such users would be motivated to take the time to learn its methods.

Data compaction was Hewlett's (1996) approach to retrieval optimization. He encoded his pitch-name data using a base-40 numbering system. This allowed him to

store most combinations of pitch and duration in two bytes.[17] He was able to store 1.5 million pitches and 100,000 rests from 357 polyphonic works by J. S. Bach in 3.2 megabytes. Because the 3.2 megabytes fit into the memory of his Pentium 100 MHz computer, he was able to perform a linear search of this dataset for a four-note monophonic pattern (B^b -A-C-H) in less than 2 seconds. However, compaction only helped so much. Another search for all the transpositions of the original pattern took 35 minutes to run. Notwithstanding that this time also includes some rudimentary statistical processing of the results (mostly counts), the difference in retrieval times is noteworthy. The second search would most definitely have benefited from some kind of intervallic indexing.

MAPPET (Music Analysis Package for Ethnomusicology) is another collection of programmes designed to assist in the encoding, retrieval and analysis of monophonic music (Schaffrath 1992a). The Essen Associative Code (*ESAC*) is used to represent the melodies. *ESAC* is a simple alpha-numeric scheme containing pitch and duration information. Melodies are first manually parsed into their constituent phrases.[18] These phrases are then *ESAC* encoded. Each encoded phrase is placed Aon its own line in one field of a relational (*AskSam*) database" (Schaffrath 1992b). There are fields containing title, key, meter, and text information. Other fields include those derived from the melodic information, such as mode, pitch profiles, and rhythmic profiles, etc. (Hewlett and Selfridge-Field 1991). The Essen databases of *ESAC* encoded melodies are the primary source for the McNab collection (McNab et al. 1996, 1997). The McNab collection was used for our own evaluations. *MAPPET*=s *ANA*(lysis) and *PAT*(tern) software sub-components can be used to translate an analyst=s complex search criteria (e.g., intervallic, scale degrees, rhythmic patterns, and so on) into *AskSam* queries. Detailed explanations of *MAPPET* and its use in the retrieval of monophonic information

---

[17]All the pitch names found in an octave, including the redundant namings (e.g., $E^b = D^\# = F^{bb}$), can be individually represented in the base-40 number system. Octave differences are denoted by multiples of 40.

[18]Phrase determination in vocal music is not ambiguous so this process is relatively easy and consistent (Schaffrath 1992a).

can be found in Schaffrath (1992b). Camilleri (1992) used *MAPPET* to analyze the melodic structures of the *Lieder* of Karl Collan.

Other examples of the many researcher toolkits available include *MODE* (Musical Object Development System) (Pope 1992), the *LIM* Intelligent Music Workstation (Haus 1994), and *Apollo* (Pool 1996).

Some researchers have seen the development of *Analytic/Production MIR* systems as a problem of programming language development. Kassler (1966, 1970) developed the Music Information Retrieval (*MIR*) language to analyze the works of Josquin des Prez. Sutton (1988) developed a PROLOG-based language called *MIRA* (Music Information Retrieval and Analysis) to analyze Primitive Baptist hymns. A PASCAL-like language was developed by Prather and Elliot (1988) called *SML* (Structured Music Language). McLane (1996) reports, however, that none of these languages has found general acceptance. McLane provides an explanation for this when he quotes Sutton: AThe literature seems to show...that scholars interested in specific musical topics have found it more useful to develop their own systems" (Sutton 1988).

3.1.2 Locating MIR systems

The *Répetoire International des Sources Musicales,* Series A/II, *Music Manuscripts after 1600* database is the official title of what is generally known as the *RISM* database. The *RISM* database is one of the oldest and "by far the most ambitious" of all MIR systems (McLean 1988; Howard and Schlichte 1988). It is an automated thematic index of gargantuan proportions. Originally conceived in the late 1940's as an attempt to catalogue over 1.5 million works, the *RISM* developers were quick to realize the need for automation (Howard and Schlichte 1988). Now in its fourth edition, the database contains bibliographic records for over 200,000 compositions by 8000+ composers (*RISM* 1997). The *RISM* database is available on CD-ROM and via the Internet at *http://www.RISM.harvard.edu/RISM/Welcome.html*. The number of indexed access points is remarkable (Figure 3-1). Of these, the **Music Incipit** index interests us the most. The music *incipits* in the *RISM* database contain pitch and duration information. *Incipits* are encoded using Brook=s alpha-numeric *Plaine and Easie Code* (Brook and Gould 1964). This is a very simple encoding scheme originally designed for

use on typewriters. Figure 3-2, taken from Howard and Schlichte=s (1988) series of examples, illustrates how *incipit* information is represented in the *RISM* database. By comparing the *Plaine and Easie* coding (a), with the Staff notation (c), one can see how pitch is denoted alphabetically, and duration numerically. The meaning of the other symbols does not concern us now.[19] The Meta-code (b) is used to generate the Staff notation (c), and is not searchable.

Names	Corporate Phrase	Library Siglum
Title Word	Call Number	Liturgical Feast
Subject Word	Content	**Music Incipit**
Names, Exact	Date of MS	Place Name
Title Phrase	Form/Genre	Publication Date
Subject Phrase	Holograph MSS	Role Name
Corporate Name	ID Number	

Figure 3-1. *RISM* database indexes

The ability afforded by the *RISM* database to search the incipits moved Amusic bibliography into a new realm" (Duggan 1992). There are, however, significant problems with accessing the *incipit* information found within the *RISM* database. First, the incipits are entered into the MARC records exactly as shown. This means that each incipit is indexed as one long, rather incomprehensible, Aword." Second, because of the way the incipit is represented in the index, queries must also be posed using *Plaine and Easie*. Third, bringing together works that contain the same melody transposed into different keys is impossible because exact pitch names are used, not intervals. Fourth, searching on pitch or rhythm exclusively is impossible for one would have to know exactly which values to wildcard along with their exact locations. Fifth, and finally, there are several, equally valid, ways to represent an *incipit*. This puts the onus on the user to frame their melodic queries in multiple ways (*RISM* 1997).

---

[19]Duggan (1992) justifiably quips that *Plaine and Easie*, A...despite its name, is not at all easy for the novice to use." We might quip that it is not very *Plaine* either.

Figure 3-2. Encoding examples, *RISM* database. Source: Howard and Schlichte (1988).

Despite its drawbacks, the *RISM* database has been the model for other projects. One such project is the development of an electronic catalogue of Sweden=s Duben Collection reported by Snyder (1992). It differs from the *RISM* database in that it uses *Paradox* as its database management system. Linked tables include bibliographic information about the works and their musical characteristics, including *incipits*.

David Huron and Andreas Kornstädt have recently mounted an automated thematic catalogue they call *Themefinder*, access to which is available via *http://musedata.stanford.edu/databases/themefinder/index.html*. The underlying retrieval engine is Huron's *Humdrum* system with Common Gateway Interface (CGI) scripting coded by Kornstädt. There are 8500 items represented within *Themefinder*, however, public access is currently limited to the works of Beethoven because of unresolved copyright issues (Huron 1999). Given the inherent power of the *Humdrum* system, the number of access points and query modes afforded by *Themefinder* is quite remarkable (e.g., key, pitch, contour, rhythm, style, orchestration, etc.) Again, because of *Humdrum*'s flexibility, the search syntax for *Themefinder* is somewhat complex and non-intuitive.

The advent of multimedia capabilities in the world of personal computing prompted a rising interest the development of prototype *Locating MIR* systems. Fenske (1988) briefly describes a project at OCLC, led by Jeanette Drone, called *Hyperbach, a Hypermedia Reference System*. This system is also described by Duggan (1989) as having “search access from Schmeider number and music entered through a MIDI interface and keyboard synthesisers.” This is the extent of information available about the *Hyperbach* system. Hawley (1990) also developed a limited system that uses a MIDI keyboard as the query interface to find tunes whose beginnings exactly match the queries. Ghias et al. (1995) developed a more sophisticated prototype system that tracks “...pitch using an autocorrelation method, converts it to a melodic contour, and matches the contour against a database of 183 songs” (McNab et al. 1996). The system described below is a more ambitious manifestation of the Ghias et al. approach.

Within the context of this thesis, one research project stands out as being particularly relevant: McNab et al. (1996, 1997). The 1996 paper reports upon the

development of a prototype *Locating* MIR system, called *MELDEX*, that has as a key feature, a pitch tracking interface that allows users to sing their queries. The interface, while fascinating, falls outside the domain of this thesis. What does interest us, however, are the methods used to retrieve the requested items from a database of nearly 10,000 folk songs, compiled mainly from the previously mentioned Essen collection. They analyzed six retrieval methods, each varying in the degree of representational completeness. These methods are listed in Figure 3-3 along with the number of notes required, on average, to uniquely identify a melody.

Exact matching method:

1. Interval and rhythm (approx. 5 notes)
2. Contour and rhythm (approx. 6 notes)
3. Interval (approx. 7 notes)
4. Contour (approx. 11 notes)

Approximate matching method:

6. Interval and rhythm (13 notes)
7. Contour and rhythm (17 notes)

Figure 3-3. *MELDEX* retrieval results. Source: McNab et al. (1996).

The McNab team sees melody retrieval as one of string matching. They appreciated the vagaries of melodic information and adopted the approximate string matching methodology of Mongeau and Sankoff (1990). This methodology was designed explicitly for the musicological analysis of melodic strings. For their contour searches they modeled their approach on Parsons' (1975) interval direction method (i.e., *****, **R**, **U**, **D)**. Contours and interval values were constructed on-the-fly, as theirs was a linear scanning retrieval process. Ranks were associated with the retrieval results based on the degree of similarity between query and the items returned. It is important to note that matches were limited to *incipits*, which they noted as a shortcoming.

Rudimentary informetric analyses were performed concerning query length, database size and the unique identification of melodies. They found that the number of notes needed to uniquely identify a particular song scaled logarithmically with database

size.[20] Missing from their analysis was an evaluation of their retrieval results using the familiar measures of recall and precision. Also, they seemed to focus their energies primarily on *uniquely* identifying melodies. This runs somewhat contrary to the traditional IR approach which recognizes the collocatory function of such queries as, AGive me *all* items containing the string "*WXYZ*."

The McNab team also reported on the types of errors that their ten subjects committed in the singing of their melodic queries. We have categorized the reported errors into four types: *Expansion*, *Compression*, *Repetition*, and *Omission*. Details concerning our classification of the query errors are found in Chapter 5. We have simulated the effect of query errors to test for the error tolerance of our indexing scheme and have based the method of simulation on the findings of the McNab researchers.

Of particular pertinence to this study is the rather slight difference in retrieval effectiveness between the "**Interval and rhythm**" and the "**Interval**" searches, as measured by the average number of notes required to uniquely identify the melodies. The former requires approximately five notes, and the latter approximately seven notes, on average, to uniquely identify each song. Thus, for the purposes of unique identification, the addition of rhythmic information reduces the required query length by only two notes. Another way of looking at this is to take the length of five notes as the baseline, and then compare the average sizes of the returned sets. A query of five notes using the "**Interval**" feature results in an average of four songs returned.[21] This is not an onerous retrieval set by any measure. These findings suggest that the inclusion of rhythmic information does not greatly improve retrieval effectiveness. We have taken this finding as further justification for limiting our present research to an examination of interval-only representations of melodic information.

---

[20]This finding relates well to the findings of those evaluating text retrieval systems. Heaps (1978) notes that the indexes to databases, where the frequency of term occurrences are Zipf-distributed (Zipf 1935, 1949), also grow logarithmically with database size. Term distribution modeling is an important component of this thesis (see Chapter 6).

[21]This value is interpolated from a graph found in the paper.

McNab et al. (1997) expands upon, and further develops, the work reported in McNab et al. (1996). The principal enhancement to their system was the ability to perform full-text searches. That is, their improved *MELDEX* system can now locate matches to queries where the match occurs not only in the *incipit*, but also anywhere within a melody. The results reported, essentially replicating the methods used in the 1996 paper, are impressive with simple, exact match searches, taking an average of 500 ms to perform. They did note, however, that their approximate string-matching method —the heart of their system—suffered from problems of scaling. They report (McNab et al. 1997):

> …approximate matching can take much longer. Matching a 20 note search pattern, for example, requires approximately 21 seconds. While it may be reasonable to expect a user to wait that length of time for a search to complete, much larger databases--a million folk songs, for example, or a thousand symphonies--will take an unacceptably long time to search.

In an effort to overcome the problem of scalability, the McNab team went on to modify their approximate string-matching algorithm by reducing its computational complexity. Search times were reduced but at the expense of discriminatory power. The faster, less complex, algorithm is now the default retrieval method for the *MELDEX* system.

The research conducted by Prechelt and Typke (1998) is similar to that of McNab et al. in that both teams have expended considerable effort developing query interfaces that take sound as input (e.g., whistling, singing, or humming). This audio input is then converted into the appropriate query strings. Like the *MELDEX* system, the work of Parsons (1975) figures prominently in the development of the *Tuneserver*. In fact, the *Tuneserver's* 10,370 item database ***is*** the Parsons' database in machine readable form. As such, the *Tuneserver* database is a relatively limited system where searches are constrained to *incipits* and themes, none of which are longer than 15 intervals. Matches to queries are made via a linear scan of the database file. Results to a query are ranked according to the *editing distance*[22] between the query string and the database entries.

---

[22] Editing distance is the number of deletions, insertions and replacements of individual characters needed to convert one string into another (see Ukkonen 1985).

Test searches took approximately 1 second on a 167 MHz UtlraSparc1. Given the simplicity of the database, and the fact that theirs is a linear search of the database, scaling the *Tuneserver* method to a million item, full-text collection appears to be problematic. Notwithstanding the scaling difficulties, Prechelt and Typke correctly recognized the potential of minimalist representations *à la* Parsons. When it comes to transmitting queries from a JAVA client program running on a World Wide Web browser, across the Internet, to a centrally located database, the 10 to 30 bytes needed to represent the Parsons-encoded query compare very favourably to the several hundred kilobytes of the WAV file representation.

The MIR research of Uitenbogerd and Zobel (1999) has many commendable components. They developed and evaluated an ambitious three-staged approach to the MIR problem. Stage 1 was "melody extraction," wherein they tested various extraction algorithms to reduce polyphonic music to a sequence of non-chordal notes. Stage 2 was the "standardization" procedure whereby sequences were rewritten "in a standard form that preserves the 'feel' of the melody but eliminates any performance-specfic characteristics" (Uitenbogerd and Zobel 1999).[23] Stage 3 was the application and testing of various "similarity functions."

The most noteworthy aspect of Uitenbogerd and Zobel (1999) was their decision to experimentally define a set of queries along with a set of relevant documents for each of those queries. The performance of their various combinations of methods was evaluated using the traditional precision and recall metrics. In short, Uitenbogerd and Zobel (1999) conducted their MIR experiments under the Cranfield Model.

Queries of 10, 30, and 100 notes were submitted against their music database of 10,466 MIDI encoded songs. They implemented a factorially complete design, which evaluated all the combinations of their extraction, standardization, and similarity methods. They found that:

---

[23] One of the "standardization" procedures evaluated was the *C23* classification scheme found in Downie (1995).

> …simple melody extraction (taking the highest note starting at any time), relative pitch intervals, and local alignment gave excellent effectiveness: for even short queries, the top 20 answers contained 12 correct matches on average (Uitenbogerd and Zobel 1999).

Uitenbogerd and Zobel finish their report with the statement that the creation and use of indexes would be investigated in future research iterations. Indexes, they concede, would improve system performance as their search methods all involved linear scans of the database. In addition, we note that their conclusions were based upon the descriptive data derived from their experiments. No inferential tests for statistical significance between or within experimental factors were performed.

There is a growing interest in MIR concepts and procedures. Although published after we had completed our research, *Computing in Musicology* (Hewlett and Selfridge-Field, eds. 1998) has devoted an entire volume to issues surrounding melodic similarity. For those interested, we recommend the complete volume. However, several of the articles are exemplary explorations of some of the fundamental concepts in MIR development. Selfridge-Field (1998) provides an excellent overview of the myriad problems associated with MIR development. Crawford et al. (1998) reviews the wide variety of string-matching techniques that can be used in MIR. Howard (1998) discusses an interesting procedure for sorting music incipits.

3.2 Summary

In this chapter, we have highlighted the principal methods used in the development of MIR systems. We have shown the predominance of linear scanning and regular-expression search methods in MIR systems. We have also noted the scaling difficulties associated with linear scans of melodic databases. The application of music-specific indexing (i.e., pitch indexes, intervallic indexes, etc.) has been recognized as potentially beneficial for retrieval performance but rarely implemented and never formally evaluated. Most *Analytic/Production* MIR systems are really collections of many task-specific sub-programmes. With the notable exception of McNab et al. (1997) and Uitenbogerd and Zobel (1999), access to music information in *Locating* MIR systems has been traditionally limited to *incipits* or themes.

We have also noted that at the time we conducted this research, no formal IR evaluations of MIR systems could be found in the literature. More specifically, there were no formal IR evaluations based upon the Cranfield Model. Only recently have other researchers (i.e., Uitenbogerd and Zobel 1999) independently picked up on the idea that the Cranfield model of evaluation should be the *sine qua non* of MIR system evaluation. Furthermore, there are no published accounts of inferential statistical analyses used to evaluate MIR system performance. These facts, when taken together with McLane-s (1996) comments, suggest that the research reported upon in this thesis should make an original and significant contribution to the fields of Information Science and Information Retrieval.

4 The MusiFind Music Information Retrieval Project

4.1 Introduction

The MusiFind Music Information Retrieval Project began in 1993 as an independent study by the author under the supervision of Dr. Tague-Sutcliffe. Throughout the course of the research Shane Dunne, Adam Wood-Gaine, Hemin Xiao, Kevin Kennedy, and several others have assisted the author and Dr. Tague-Sutcliffe with data processing and computational suggestions. Since the death of Dr. Tague-Sutcliffe, supervision and support of the MusiFind work has been ably undertaken by Dr. Micheal Nelson. It is to this rather ephemeral group that the author refers when mentioning the MusiFind researchers.

In this chapter, we will discuss in some detail the conceptual framework, operating paradigms, limitations, and preliminary results of the MusiFind project. This information is given in order to provide the necessary background information concerning the thesis research proper, which follows in Chapters 5-8.

4.2 The MusiFind conception of a true music database

A true music database should contain, in addition to the standard bibliographic information found in online catalogues of recordings and musical scores, some type of musical representation that allows for the searching and subsequent retrieval of the contents of the music stored within. Such a true music database would also have the following features:

Design Feature 1: An interface that has the graphical ability to handle standard musical notation both for queries and the presentation of results.

Design Feature 2: Music keyboard and microphone interfaces for played or sung queries.

Design Feature 3: Outputs that include audio, video and printing presentations of both full scores and parts.

Design Feature 4: The ability to locate both melodic and harmonic fragments anywhere in a musical work.

4.3 Operating paradigms of the MusiFind research project

To gain a fuller appreciation of the development of the MusiFind project it is important to understand the two paradigms under which its research has been conducted.

Paradigm 1: Apply a full-text, bibliographic information retrieval (FBIR) model to the retrieval of music information.

Paradigm 2: Apply the principle of parsimony to all design and evaluation decisions.

These paradigms and their implications are explained below.

4.3.1 Explication of Paradigm 1

Paradigm 1 refers to a design decision that reflects the intended uses of the MusiFind system. In the previous chapter a distinction was made between *Analytic* and *Locating* MIR systems. The MusiFind team decided that theirs would be a *Locating* MIR system. That is, they intended that their system would find broad-based use in libraries, music stores, radio stations, etc., by users striving to locate and use (i.e., listen to, print out, purchase, etc.) some musical composition(s) rather than performing musicological analyses upon them. This decision would later be supported by the research reported in Downie (1993d), which is summarized later in this chapter.

In a standard FBIR system each document is usually represented by a record that contains such bibliographic information fields as **Title**, **Author**, **Source**, **Descriptors**, **Publisher**, etc., and a field, **Full-text**, where a *"machine-accessible manifestation"* of a document resides (Dunne 1993). These fields are also known as attributes. The bibliographic attributes contain information ***about*** a document. The **Full-text** attribute, however, contains the information that the creator(s) of the document wished to communicate. Furthermore, the expression of the information is that of the creator(s), not some intermediary. Providing access to information as expressed by the creator(s) of the information is a fundamental purpose of an FBIR system.

For example, if a user wants to find all works contained in the database by a given author (say, ***Adam Smith***) the usual procedure would be to perform a search limited to the **Author** field with "***Adam Smith***" submitted as the search statement.[24] If, however, the user wants to find those documents where ***Adam Smith*** discusses his famous "***Invisible Hand***," the user would do best by combining the above **Author** search with a search of the **Full-text** field using "***Invisible Hand***" as the search statement.

In the context of music, we can substitute Bach for Smith. To access his musical expressions we would search the **Full-text** field of the system where his compositions would be stored. Just how we would store those compositions for access is discussed below.

4.3.2 Implications of Paradigm 1

There are two important implications that follow from the adoption of an FBIR model for music information retrieval. First, it is important to distinguish between the *records* in an FBIR system and the *indexes* used to locate those records. When one "searches" the **Author** field, the FBIR system does not search the **Author** fields of each record in turn. Instead an index, traditionally in the form of an inverted file, is searched where the terms found in the **Author** fields of all the records have been stored along with some type of record identification code (RIC).[25] The RIC points to the specific record(s) from which the given term(s) came. A search of the **Full-text** field is likewise conducted. If a search is successful the record(s) corresponding to the RIC(s) stored in the index(es) are retrieved from a linear file where the records are actually located. This implies that the means of representing information in the indexes and in the records need not be identical. For example, in FBIR systems the records contain capitalization and/or punctuation information that is usually omitted from the indexes.[26] Thus, in an MIR

---

[24]The actual syntax of the search statement is system dependent.

[25] An *inverted file* is a list of some attribute(s), ordered on the terms within the list (usually alphabetically) as opposed to being ordered on their locations within the collection.

[26] In systems that employ stemming, only the stem of a given term would be stored in the inverted file. This singular entry would be used to locate all the morphological variants of the term in the records.

system, the representation of the musical compositions stored in the records need not be the same as the representation used in the index(es). Therefore, the contents of a musical work might be represented within a record by a photographic image of the score, DARMS codes, a sound file, a MIDI file, or whatever, without affecting the choice of indexing method. Because of this index/record independence the MusiFind researchers decided to adopt a music indexing strategy that would be independent of the representation of the music used in the records.

Given that the MusiFind researchers decided to create a MIR system under the FBIR model, and that the indexes to the music component of the system could be independent of the representation of the music in the records, it followed that it would be advantageous to use an indexing representation that would be amenable to integration into existing FBIR systems. Since most FBIR systems index strings of alpha-numeric characters in the form of words, it also followed that the indexing representation of the music should also use alpha-numeric characters structured in a manner that would mimic the properties of words. Thus, if the indexing representation of music as "words" proved to be successful, then it might be possible to create a MIR system using one of the pre-existing FBIR software packages. For these reasons, the MusiFind project has focussed upon the possible creation of "musical words" and their potential utility in the creation of MIR system indexes.

The second implication of Paradigm 1 concerns the choice of criteria that would be used to determine whether the "musical words" approach to music indexing was well-founded. If "musical words" give the same or better access to music information than "real words" give to textual information (as reported in the IR literature) then the "musical words" approach should be deemed successful. Since the FBIR model had been adopted it followed that the criteria used to evaluate FBIR systems should be used. As stated in the previous chapter, *Precision* and *Recall* are the principal evaluative measures associated with FBIR evaluation. Thus, these measures (or some variant) would be used to evaluate the retrieval effectiveness of indexing "musical words."

4.3.3 Explication of Paradigm 2

The principle of parsimony underpinning the MusiFind research was inspired by the model comparison method of data analysis (see Judd and McClelland 1989). Simply put, it asserts that a simpler model is to be preferred over a more complex model unless a more complex model can be proven to provide significantly better results. Furthermore, the amount of improvement must be evaluated in light of the complexity added by the more complex model. Judd and McClelland (1989) denote the simpler model as Model **C** (**C** for **C**ompact) and the more complex model as Model **A** (**A** for **A**ugmented). In data analysis there are formal statistical tests that can be used to determine when Model **C** should be abandoned for Model **A.**

Another expression of the principle of parsimony is the more familiar notion of cost/benefit analysis. So, even without the assistance of formal statistical tests, the principle of parsimony can be applied heuristically. To do so, however, one must have answers to the following questions:

Q1) What are the desired benefits?

Q2) What are the baseline measures of the benefits?

Q3) What are the costs associated with increasing the benefits?

Q4) What are the criteria for determining when the increase in benefits outweighs the costs?

There is a great deal of debate in the IR literature over the appropriate choice of benefits, measures, costs and criteria (Tague-Sutcliffe 1992*;* Keen 1992; Harter and Hert 1997). The matter is complicated by the fact that those things defined as benefits and those as costs can be interchanged depending on the individual preferences of a researcher. For example, one investigator might consider an increase in precision the desired benefit (with the possible concomitant cost of processing time) while another might consider a decrease in processing time the desired benefit (with the possible concomitant decrease in precision (e.g., McNab et al. 1997)). Further complications arise in the interactions of desired benefits, that is to say, some sets of desired benefits can be simultaneously considered as costs. The classic example of this interaction is the

persistent trade-off between precision and recall (Harter and Hert 1997). In general, however, the following answers to the preceding questions have found considerable use in IR system design and evaluation:

A1) Increases in precision, recall, and user satisfaction are the most commonly desired benefits along with decreases in storage requirements, processing times, search times, and monetary expenses.

A2) Baselines can be established by reference to previously published research or by conducting controlled experiments.

A3) Increases in storage requirements, processing times, search times, and monetary expenses, as well as decreases in user satisfaction, precision and recall can all be considered costs.

A4) Sparck-Jones' "material difference," usually set at 10%, can be used as the criterion for preferring one design or system over another (Sparck-Jones 1974; Keen 1992; Harman 1995)

To conclude, the principle of parsimony is a very useful decision-making aid. If one has a clear idea of the desired benefits and an appreciation of the costs of those benefits, it makes deciding between competing options much less onerous. It also implies that research should begin with:

1) the exploration of pre-existing methods; and,

2) the simplest possible representation of the data.

Both of these can be considered "simpler" solutions. It further implies that designers should continue to use only simpler solutions until such time that it can be clearly demonstrated that superior solutions exist.

4.3.4 Implications of Paradigm 2

The principle of parsimony has been used to justify all of the MusiFind project's design and evaluation decisions. Examples of just three of the many significant decisions made during the course of the MusiFind project are given below to illustrate how the principle has been applied:

Decision 1: Index only monophonic melodies

Desired benefits: conceptual simplicity, smaller indexes, smaller storage, faster searching

Major costs: loss of polyphonic information, some manual editing might be required

Decision 2: Use only the intervallic information in the creation of the indexes

Desired benefits: smaller indexes, smaller storage, faster searching, increased recall

Major costs: loss of rhythmic and editorial information, decreased precision

--This represented the use of a pre-existing solution in that Parsons (1975) and others had used this simplistic method for printed *incipit* indexes

Decision 3: N-gram[27] melodies into "musical words"

Desired benefits: allow use of pre-existing FBIR indexing and searching methods, give access to information located anywhere in the melody

Major costs: some extra preprocessing time, increase in index size

To summarize, the principle of parsimony has been applied to all of the MusiFind project's design and evaluation decisions. In other words, the researchers have constantly weighed the costs and benefits of each design decision. In general, it has led the researchers to explore what they considered to be the simpler solutions to MIR system

---

[27]Procedure for n-gramming is formally described in Chapter 4.4.1.1.

development. It is important to emphasize, however, that the author and his associates did not reject other, possibly more optimal, approaches; but, rather, the principle of parsimony implied that the simpler solutions they had chosen should be implemented and evaluated until such time as the simpler solutions could be proven not to perform adequately.

4.4 Summaries of key MusiFind reports and publications

In this section we present summaries of the principal MusiFind reports, presentations, and conference papers. The summaries below are presented in chronological order. Understanding the evolution of our thinking concerning the MIR problem and its potential solutions is central to appreciating the reasoning behind the structure and goals of the thesis research proper.

4.4.1 Creating the ideal full-text music database (Downie 1993b)

This research report presented the preliminary findings and recommendations of the MusiFind researchers. In it are discussed the strengths and weaknesses of printed thematic catalogues, *incipits* indexes, and the various musical representations amenable to computer manipulation. It concluded that any MIR system developed had to meet, if not exceed, A. Hyatt King's (1954) suggestions for the ideal thematic catalogue. The MusiFind conception of a true music database was founded upon this conclusion.

MIDI files were chosen as the encoding medium because they were readily available and the MIDI format was understood by the researchers. MIDI files also offered "quick and dirty" solutions to Design Features 1 through 3 for there were pre-existing software packages that could be used either "off-the-shelf" or slightly modified which utilized the MIDI format. Loss of editorial information inherent in the MIDI format was noted but left to another day for resolution.

Since MIDI files offered a potential solution to the first three Design Features, attention was focussed upon Design Feature 4. The use of Directed Acyclic Graphs was presented as a possible solution to the problems associated with accessing polyphonic music information but not developed fully as the researchers felt that the research should first thoroughly examine the simpler case of monophonic music information. Operating from a state of ignorance concerning the true informetric properties of monophonic

melodic information, the researchers decided that the indexing approach should begin by emphasizing recall over precision. This emphasis on recall also led to the decision not to include rhythmic information in the indexes.

It also led to the use of intervals, rather than pitches, as the primary elements in the indexing procedure because intervals are a more general representation than pitches. Thus, the author and the MusiFind team hypothesized ***that there would be enough information retained within a simple, monophonic, interval-only representation of musical works to achieve reasonable retrieval performance***. The original indexing and query procedures, formally proposed by Dunne and Downie, are presented below as they are found in Downie (1993b).

4.4.1.1 The "MusiFind Approach" to music indexing: definitions and procedures

"interval" = the signed difference between 2 pitches

If the pitches are encoded according to MIDI codes (0-127), then the intervals are subsequently encoded as -127 to +127. This presents far too many possible intervals. We must, therefore, limit our possibilities by creating "interval classes".

"interval classes" = equivalence classes of intervals

We can partition the set of intervals I *into disjoint subsets* $I_1, I_2,...I_c$ *such that* $\bigcup_{j=1}^{C} I_j = I$ *and* $I_i \cap I_j \neq 0 \rightarrow i = j$ (or $i \neq j \rightarrow I_i \cap I_j = 0$). *Each* I_i *is called an equivalence or interval class.*

Next, we define a "classification function":

"classification function" = $C : I \rightarrow [1..c] = \{1,2,3...,c\}$ such that

$i \quad I$, $C(i) = j$ where $i \quad I_j$.

This converts an interval i *into its appropriate class number.*

Now that we have dealt with concepts of individual pitches and intervals (and their corresponding incorporation into classes), we must move on to describe how we intend to deal with the concept melodic strings.

Given an ordered list or sequence of pitches $p_1,p_2,p_3,...p_n$, define the corresponding "interval sequence" as $i_1,i_2,i_3,...i_{n-1}$, such that i_j = interval between p_j and p_{j+1} (i.e. [1...n-1]).

For MIDI data $i_j = p_{j+1} - p_j$.

Given an interval sequence i_1, i_2, $i_3...i_{n-1}$, define the corresponding "interval-class sequence" as $C(i_1)$, $C(i_2)$, $C(i_3)...$, (i_{n-1}).

*This sequence has length n-1, so we call it a "*length-(n-1)*" sequence.*

Given any sequence $X = x_1,x_2,x_3,...,x_n$, the "length-k subsequence beginning at element j" is defined as the sequence $x_j,...,x_{j+k-1}$. Denote this as X^k_j.

Thus, given an interval-class sequence $S = c_1,c_2,c_3,...,c_n$, *the length-k subsequence beginning at element* j *is* S^K_j. *If* $n-k+1 > 0$ *there are* $n-k+1$ *such sequences, otherwise there are none.*

It is possible to summarize these steps in the form of an algorithm.

4.4.1.2 Algorithmic summary

1. Get a melody = pitch sequence of length n+1 (MIDI pitch numbers).
2. Take 1st-order forward differences to get the corresponding interval sequence, which has length n.
3. Apply classification function *C* to each interval, yielding corresponding interval-class sequence, also of length n.
4. Extract all length-k sequences (k is chosen arbitrarily) subsequences.

There are n-k+1 such subsequences unless n-k+1 is less than, or equal to, 0, then there are none. Also, these subsequences are not necessarily distinct.

The classification function C is completely determined by our choice of partition on the interval set (i.e. how we define the equivalency classes of intervals).

Each partition determines a particular value of c. *Note, however, that it is possible to define 2 different partitions which happen to have the same number of classes and therefore the same* c.

Given *k* and *c* (2 numbers), the total number of different length-k interval-class sequences is c^k.

So, if we wanted to build a table of all length-k interval-class sequences, that table would have c^k *entries. We then could number them* 0 *through* c^k-1.

Given a length-k interval-class sequence $S = s_0, s_1, ..., s_{k-1}$, define a "hash function"as:

$$H(S) = \sum_{i=0}^{k-1} c^i s_i$$

This will generate numbers in the range 0 *to* c^k-1. We can use these numbers to index the table, which is called a "hash table." Also, we can now add another step to our algorithmic summary.

4.4.1.3 Algorithmic summary (continued)

5. For all distinct length-k interval-class sequences in step 4, use the hash function H(S) to generate the corresponding indexes into the hash table (one per sequence).

The following is the algorithm to create the hash table (index) for a set of monophonic melodies.

4.4.1.4 Hash table algorithm

1. Initialize table so all entries are the empty list.
2. For all files (melodies) f do

   For all length-k interval-class sequences in f do

   2a. Compute the hash index i by applying H(S) to the sequence.

2b. Add the filename f (or some identifying code) to the list at the i^{th} entry of the hash table, unless f already appears in that list. *(The list actually represents a set.)*

Using the same principles used to construct our hash table, we can create query strings. Given a melodic fragment (pitch sequence) m_q *(a mnemonic for "melody query"), the following algorithm will yield the list of files which* ***may*** *contain the melody* m_q.

4.4.1.5 Query algorithm

1. Create all length-k interval-class sequences in m_q.
2. For each such sequence do
    2a. Generate the corresponding hash-table index i using H
    2b. Extract the list of files found at the i^{th} entry of the hash table.
3. Merge (AND) all the resulting lists.

The retrieved files each contain at least 1 instance of each of the length-k interval-class sequences found in m_q, *but not necessarily* m_q *itself because the sequences might not be in the proper order. Because of this, there is the possibility of numerous false drops. However, recall should be 100 percent.*

4.4.2 Implications of the MusiFind approach to music indexing

The preceding indexing and query scheme created important ramifications for the research that would follow.

First, this scheme defined interval-class sequences as analogues to words in a bibliographic database. More precisely, a length-k interval-class sequence is the analogue of a textual n-gram. Thus, it formalized a method for the creation of “musical words” through an n-gramming process.

Second, there are two arbitrary elements within the indexing scheme: *c* and *k*, where *c* = the number of equivalence classes and where *k* = the length of the indexing entry. The researchers were not aware of a theoretical basis for choosing any particular value for these variables. They conjectured, however, that some values of *c* and *k* would

yield unwanted results. For example, if they set $c = 1$, then every interval would be equal and the resulting index would be useless. Finding the proper value of k was important because both extreme brevity (i.e., $k = 1$) and extreme length (say, $k > 10$) would yield unmanageable results. Examination of various combinations of c and k would become a principal purpose of subsequent MusiFind research.

Third, the proposed hash function H(S) is a perfect hash function in that no two distinct keys would map to the same address. An advantage of this would be very impressive search times as hash table retrieval has a time complexity of $O(1)$. Use of this hash function, however, would impose important constraints on subsequent research**.** The number of buckets required by the hash table is determined by c^k. The researchers decided it would be wise then to constrain the values of c and k such that $c^k < 10^6$ so the number of buckets required would fit comfortably into the RAM of a modestly-equipped PC-class computer of 1993.

Fourth, the potential for query-error forgiveness was created. To illustrate, let us compare two possible values of c: $c_a = 3$ and $c_b = 23$. For c=3, let us use the Same-Up-Down (*SUD*) scheme where, n(C_a) represents the number of elements in a set of interval-class equivalencies C_a such that n(C_a) = 3 and C_a = {c_{ax}| x 3, x *W*, c_{a1}={any zero interval}, c_{a2}={any interval upwards} and c_{a3}={any interval downwards}}. For $c_b = 23$ let us define the classification function as $C_b(i) = i \bmod 12$, making n(C_b) = 23, where i is the signed interval taken from the melody. Now consider two melodic queries, mq_1=0 7 0 2 0 , and mq_2= 0 7 0 1 0, where the melodies are represented by their unclassified intervals. Using the C_a classification both mq_1 and mq_2 resolve to $mq_{ca(1|2)}$ = 0 1 0 1 0. C_b classifies these as mq_{cb1}=0 7 0 2 0 and mq_{cb2}= 0 7 0 1 0. Under some circumstances it is possible to consider mq_2 as being a malformed version of mq_1 where the penultimate interval (1) in mq_2 has replaced the proper (2) in mq_1. If this is the case then it is readily apparent that C_a has afforded the user a degree of error forgiveness not provided by C_b as C_a defined the two queries as being equivalent. In general, the degree of forgiveness is inversely proportional to the value chosen for c. A corollary of this is that the degree of precision afforded by a classification scheme is proportional to the value selected for c; that is, precision increases as c increases. While the forgiveness/precision trade-off has always been known to the author and his associates, it would not be examined in any

detail. It would, however, re-emerge as an item of interest in the experiments outlined later in this thesis.

4.4.3 Name that tune: an introduction to music information retrieval (Tague-Sutcliffe et al. 1993)

In this conference paper Tague-Sutcliffe et al. codified the findings presented in Downie (1993b). It extended those findings by comparing and contrasting the relative merits (and complexity analyses) of four possible methods of accessing melodic information. These methods were:

1) Linear scanning $O(n)$
2) Inverted files $O(\log n)$
3) Perfect Hashing $O(1)$
4) Bit-vector $O(n)$[28]

Method 1 was presented only to establish a baseline measure against which the other methods would be compared. Discussions of methods 2 through 4 were all predicated on the use of the length-k interval-class sequences. The authors noted that the potential usefulness of methods 3 and 4 (and to a lesser extent method 2) would be strongly determined by the values selected for *c* and *k*. The paper concluded with the suggestion that empirical investigations of various combinations of *c* and *k* should be conducted.

4.4.4 Creating the ideal full-text music database: user assessment survey (Downie 1993d)

In addition to the aforementioned discussions of indexing methods, both Downie (1993d) and Tague-Sutcliffe et al. (1993) called for an examination of user needs and perceptions as they related to music information retrieval. In the summer of 1993 Downie conducted an independent study under Tague-Sutcliffe designed to elicit this information. A videotaped demonstration of a MIR system mock-up was presented at

---

[28]Bit-vector and linear searches both have $O(n)$ time complexity. However, searches of bit-vector indexes are much faster than linear searches because bit-vector indexes take advantage of a computer's ability to perform bit-wise operations very efficiently.

five lecture/demonstrations: one at IBM in Don Mills, Ontario, two at the Graduate School of Library and Information Science of the University of Western Ontario, one at the Faculty of Music of the University of Western Ontario, and one at the London Public Library of London, Ontario. In all, 41 of the attendees, representing a wide range of music skills and tastes, provided both qualitative and quantitative data. Six of the principal conclusions drawn from the analysis of the data are presented below:

1) There seemed to be both a need for, and a general acceptance of, the features defined in the MusiFind conception of the ideal MIR system.

2) Practicing librarians seemed particularly enthused by the prospect of using our proposed MIR system as a reference source.

3) The preferred query methods were (in order): "Title," "Composer," "Lyric Text," "Singing," "Music Style," and "Music Keyboard." Also the users wished to be able to combine various search methods.

4) Performance ability did not seem to have an effect on the users' desire to exploit the "Singing" and "Keyboard" searches.

5) There did, however, appear to be a relationship between the users' ability to read standard music notation and the acceptance of the "Notation" search.[29]

6) The multimedia output options were not as enthusiastically supported as the various query methods, suggesting that the users wished to use the proposed MIR system as a locating device for external music sources (i.e., recordings).

Conclusions 2, 5, and 6, in particular, gave strong support to the decision to develop a *Locating* rather than an *Analytic* MIR system (Paradigm 1). Thus, the results of this study led the author and his associates to conclude that the MusiFind conception of a MIR system was well-grounded.

---

[29]"Notation" searches were those where the queries would be represented using the standard graphic note-and-staff representation.

4.4.4.1 The MusiFind music information retrieval project, phase II: user assessment survey (Downie 1994)

This paper represents the published version of Downie (1993d). It was written for presentation at the 1994 conference of the *Canadian Association for Information Science*.

4.4.4.2 The MusiFind music information retrieval project, phase III: evaluation of indexing options (Downie 1995)

The purpose of this study was to examine the retrieval effectiveness of n-gramming interval-only melodic strings using the MusiFind indexing procedure outlined in Chapter 4.4.1.1. The melodic strings were indexed from a database of 100 MIDI encoded monophonic songs. Music styles ranged from Negro spirituals (e.g., *Swing Low*) to Baroque (e.g., *Pachebel's Canon*) to sea shanties (e.g., *Blow the Man Down*). Each piece was approximately two to four minutes long and contained between 80 and 200 notes. It was hoped that the broad selection of musical styles would help the author in determining which combinations of *c* and *k* would be best for all types of music, where *c* was the number of elements in the classification scheme and *k* the length of the interval-class sequence (i.e., n-gram or "musical word").

Five interval classification schemes (*C3*[*SUD*], *C7*, *C12*, *C14*, and *C23)* and four index segment lengths (*L3*, *L4*, *L5*, and *L6*) were evaluated, where *Cx* is the size of the classificatory set and *Lx* is the length of the n-gram in intervals. Twenty databases were therefore constructed and the number of unique terms found within each were analyzed. These combinations were also evaluated with regard to the theoretical maximum number of unique index entries each could produce as determined by the equation c^k. These maxima are given in Table 4-1.

Table 4-1. Theoretical maxima of unique index terms

	SUD	C7	C12	C14	C23
L3	27	343	1,728	3,744	12,167
L4	81	2,401	20,736	38,416	*379,841*
L5	234	16,807	*248,832*	*537,824*	**6,436,343**
L6	729	*117,649*	**2,985,98**	**7,529,536**	**148,035,889**

The bolded combinations were initially rejected from consideration because the resulting size of their hash table indexes would be too large for storage within the then-

current PC-class computer. The italicized cells in Table 4-1 indicate those combinations selected for later experimentation because their theoretical maxima fell within the 10^5 to 10^6 range. This range was conjectured to be better suited for use in indexing larger collections at some future date. Table 4-2 presents the configurations of the five classification schemes *C* used in the study.

Informetric analysis revealed that most of the combinations produced highly-skewed, Zipf-like frequency distributions (Zipf 1935) as measured by the number of songs in which each unique term was found. That is to say, that the number of terms that occurred only once was very high while the frequency of terms with multiple postings quickly diminished as the number of postings increased. Exceptions to this were *SUDL3* and *SUDL4* distributions which were fairly uniform and *SUDL5* and *SUDL6* which could be classed as negative binomial distributions. *C14L4*, *C14L5* and *C23L4* were extremely skewed with the vast majority of terms occurring only once.

4.4.4.3 Retrieval experiments

Four combinations of indexing length and classification were chosen for experimentation: *C7L6*, *C12L5*, *C14L5* and *C23L4*. The indexes contained no within-song location information. Query fragments (*Q)* of length-4, length-6, and length-8 were extracted from the *incipits* of each song. These lengths were believed to represent the majority of query lengths of potential users. In total 1,200 queries were run (100 songs X 4 databases X 3 query lengths = 1200). When the query length exceeded the index term length the query was fragmented into the appropriate length-*L* segments and all length-*L* segments were run. For example, a query of **12345** when presented to a *L4* database was entered as **1234** *and* **2345.** The set of hits were the intersection of the resulting posting lists. For queries that were shorter than the index entry length, the queries were "wildcarded" to the appropriate length. For example, a query of **1234** presented to a *L6* database was entered as **1234@@.** The set of hits was the union of the resulting posting lists.

The retrieval effectiveness of the four chosen combinations was evaluated using a modified measure of precision. Precision was defined as **P =1/number song titles retrieved**. Thus, for the purposes of this study, a non-relevant hit was any song title

retrieved other than that from which the query was extracted. This measure was considered adequate as the goal of this stage of research was to quickly identify those combinations of length *L* and classification scheme *C* that behaved pathologically. The results are summarized in Table 4-3.

Table 4-2. Classification schemes

Codes	Classification Scheme	Commentary
SUD (C3)		
-1	Any interval downwards	- Least discriminating scheme - Most forgiving scheme for inaccurate queries
0	Note repeats	
1	Any interval upwards	
C7		
0	Note repeats	-Grouped by general intervals (i.e. 2nds, 3rds Perfects, etc.) -Direction irrelevant, - Still a rather forgiving scheme
1	Unsigned 1 or 2	
2	Unsigned 3 or 4	
3	Unsigned 5 or 7	
4	Unsigned 8 or 9	
5	Unsigned 10 or 11	
6	Unsigned 6	
C12		
0	Note repeats(any octave)	- Grouped by semitones - Direction irrelevant - Substantial decrease in forgiveness
1	Unsigned 1	
2	Unsigned 2	
3	Unsigned 3	
4	Unsigned 4	
5	Unsigned 5	
6	Unsigned 6	
7	Unsigned 7	
8	Unsigned 8	
9	Unsigned 9	
10	Unsigned 10	
11	Unsigned 11	

Table 4-2. Classification schemes (Continued)

Codes	Classification Scheme	Commentary
C14		
0	Note Repeats	-Grouped by general intervals (as in C7) but direction is relevant -Nice mix of discrimination (direction) and forgiveness (general intervals)
1	+12	
2	+1 or + 2	
3	+3 or +4	
4	+5 or +7	
5	+8 or +9	
6	+6 or –6	
7	+10 or +11	
8	-10 or –11	
9	-8 or –9	
10	-5 or –7	
11	-3 or –4	
12	-2 or –1	
13	-12	

C23				
Code	**Scheme**	**Code**	**Scheme**	**Commentary**
0	Note repeats (any octave)			-Most discriminating -Least forgiving (i.e., queries have to be exact)
1	+1	12	-11	
2	+2	13	-10	
3	+3	14	-9	
4	+4	15	-8	
5	+5	16	-7	
6	+6	17	-6	
7	+7	18	-5	
8	+8	19	-4	
9	+9	20	-3	
10	+10	21	-2	
11	+11	22	-1	

The Table 4-3 headings require explanation. The column **SD** shows the standard deviation of the mean number of titles retrieved for each combination of classification scheme (*Cx*), indexing length (*Lx)* and query length (*Qx)*. The **Max** column reports the largest retrieved set; that the **Max** for *C7L6Q4* was 83 indicates that one query retrieved

83 titles. The **Mean** column shows the mean number of retrieved titles averaged over the 100 queries. The **Unique Queries** column shows the number of unique queries generated; when this value is less than 100, it indicates that some combinations generated duplicate queries from different song beginnings. **P** is the precision measure discussed above. Finally, **Mean P** is the average **P** value for each classification-scheme/index-length combination.

Table 4-3. Retrieval results (*incipit* queries)

	SD	Max	Mean	Unique Queries	P	Mean P
C7L6Q4	19.64	83	24.93	69	0.04	0.14
C7L6Q6	9.96	53	7.8	87	0.13	
C7L6Q8	7.21	53	3.84	97	0.26	
C12L5Q4	10.56	42	9.95	87	0.10	0.40
C12L5Q6	3.09	16	2.6	97	0.38	
C12L5Q8	1.58	14	1.41	99	0.71	
C14L5Q4	8.79	39	8.34	90	0.12	0.43
C14L5Q6	2.63	17	2.26	96	0.44	
C14L5Q8	1.47	11	1.35	100	0.74	
C23L4Q4	5.06	26	4.86	93	0.21	0.50
C23L4Q6	2.8	26	1.84	97	0.54	
C23L4Q8	2.54	26	1.35	100	0.74	

4.4.4.4 Observations and conclusions

As the size of the classification scheme *C* increased so did precision (**P**). **P** also increased as query length *Q* increased. These results were not surprising as an increase in both *Q* and *C* would obviously bring about an increase in discriminatory power. Regardless of query length, precision increased substantially as *C* increased.[30] Note that index length *L* seemed to be inversely proportional to precision.

The results of these evaluatory trials were very encouraging. These results suggested that a reasonable precision rate was possible without the need to include

[30]The reader is reminded however that these results reflected only one set of possible combinations of *C* and *k* and that any size *C* can take on many different configurations.

within-song term locations. This being the case, n-gramming the interval-only representations of melodic strings (i.e., the creation of "musical words") remained a workable approach. The data also suggested that the interactions between *C*, *L*, and *Q* would require further study as the simple descriptive statistics gathered could not accurately describe the true nature of these potentially important interactions.

4.4.5 Colloquium: Toward the creation of a full-text music information retrieval system: a presentation of findings and future directions (Downie 1996b)

Two questions arose after the author conducted the research summarized above. First, would query strings that represented melodic strings located somewhere other than a song's *incipit* be as effective as *incipit* strings? And second, what kind of results could one expect using an exact string search where both the melodies and queries were represented as contiguous strings of unclassified intervals (i.e., do not apply the MusiFind approach to indexing)?

To answer the first question the author reused the four databases examined in the earlier study. Query lengths used were again *Q4*, *Q6* and *Q8*. Queries were created by extracting strings of the appropriate length from randomly selected starting points in each of the 100 songs. Thus, 100 *RQ4*, 100 *RQ6* and 100 *RQ8* queries were extracted, where *RQx* indicates that the query is a random location query of length-*x*. The method of the preceding study was then replicated. The results are summarized in Table 4-4.

To answer the second question, the author reused the 100 contiguous strings of unclassified intervals used to create the n-grammed databases. The *incipit* strings used in the preceding study were reused to represent *incipit* queries (*IConQx*). The contiguous strings created above were used to represent random location queries (*RConQx*). A simple exact string-matching programme was written in the ObjectPal language. The results of the exact string-matching tests are found in Table 4-5.

4.4.5.1 Observations and conclusions

Averaged over all conditions, the n-grammed random location queries showed a slight improvement in precision (mean **P** = 0.40) compared to the n-grammed *incipit* queries (mean **P** = 0.37). The grand mean precision for all n-grammed databases (both

random and *incipit* queries of all lengths) was 0.38 while the grand mean for the contiguous queries (both random and *incipit* queries of all lengths) was 0.63. The superiority of the contiguous string searches was not unexpected. Of the four databases the *C23L4* database, which had a mean **P** = 0.53, averaged over both random and *incipit* queries of all lengths, came closest to the effectiveness of the contiguous string searches (mean **P** = 0.63).

A more telling comparison involved the mean number of documents returned for each query length (see the **Mean** columns in Tables 4-3, 4-4, and 4-5). Subtracting 1 from each value in the **Mean** columns yielded the number of songs retrieved other than the ones from which the queries were extracted. The author called these songs, "extra songs." The mean number of "extra songs" retrieved from the *C23L4* database was 1.60, averaged over both *incipit* and random queries of all lengths. The mean number of "extra songs" retrieved by the contiguous string searches was 1.43, averaged over both *incipit* and random queries of all lengths. Given that the 1.43 "extra songs" represented songs within which the query strings were found intact, the extra songs retrieved by the contiguous string searches were all "legitimate responses" to the queries. Thus, subtracting the mean number of "extra songs" retrieved by the contiguous string searches (1.43) from the mean number of "extra songs" retrieved from the *C23L4* database (1.60) gave the mean number of true false drops per search (1.60 - 1.43 = 0.17). These 0.17 false drops were solely attributable to the lack of within-song location information within the *C23L4* index. This meant that, on average, a search of the *C23L4* database retrieved substantially less than 1 song per query that could not be considered a "legitimate response" to the query.

Applying the principle of parsimony (Paradigm 2) to this finding led the author to conclude that the benefit created by the linear searches of the contiguous strings (i.e., no false drops) was not offset by a great enough margin (i.e., the average 0.17 false drops per query caused by the lack of within-song location information) to justify the $O(n)$ time complexity of the linear search as compared to the time complexities of $O(\log n)$ (binary searches) or $O(1)$ (hash table searches) afforded by the MusiFind approach. Furthermore, the author conjectured that the increase in index size needed to include within-song location information would not justify the elimination of 0.17 false drops per query.

Table 4-4. Retrieval results (random location queries)

	SD	Max	Mean	Unique Queries	P	Mean P
C7L6Q4	20.48	83.00	23.19	75.00	0.04	0.14
C7L6Q6	10.22	53.00	7.57	92.00	0.13	
C7L6Q8	7.83	53.00	4.15	95.00	0.24	
C12L5Q4	11.94	42.00	11.68	79.00	0.09	0.43
C12L5Q6	2.47	14.00	2.13	98.00	0.47	
C12L5Q8	1.44	14.00	1.36	100.00	0.74	
C14L5Q4	9.61	39.00	8.84	87.00	0.11	0.47
C14L5Q6	2.51	15.00	2.04	99.00	0.49	
C14L5Q8	1.00	8.00	1.24	100.00	0.81	
C23L4Q4	5.25	26.00	4.81	95.00	0.21	0.55
C23L4Q6	0.95	6.00	1.43	100.00	0.70	
C23L4Q8	2.53	26.00	1.35	100.00	0.74	

Table 4-5. Retrieval results (contiguous string searches, *incipit* and random location queries)

	SD	Max	Mean	Unique Queries	P	Mean P
IConQ4	4.75	22.00	4.59	90.00	0.22	0.62
IConQ6	1.1.7	8.00	1.41	96.00	0.71	
IConQ8	0.44	5.00	1.08	99.00	0.93	
RConQ4	5.78	26.00	5.17	397.00	0.19	0.64
RConQ6	0.93	8.00	1.30	396.00	0.77	
RConQ8	0.32	5.00	1.04	396.00	0.93	

To summarize, there was very little difference in retrieval effectiveness between *incipit* and random queries. Overall, contiguous string searches performed better than searches of the n-gram databases. However, the number of false drops created by the lack of within-song information did not appear great enough to abandon the simple n-gram indexes for a more precise linear search (i.e., the application of the principle of parsimony).

4.4.6 Concluding remarks on the MusiFind project

The cumulative findings of the MusiFind project were strongly suggestive that the MusiFind approach of n-gramming of interval-only melodic strings into "musical words" was a useful strategy for accessing music information. It must be stressed, however, that these findings were only suggestive and not conclusive. They are only suggestive because:

1) The database size of only 100 songs was too small to provide generalizable results;
2) not all combinations of n-gram length *L* and classification scheme *C* were evaluated making it impossible to clarify the true effects caused by varying *L* and *C;* and,
3) no formal tests were applied to determine if the differences in retrieval effectiveness observed could be considered statistically significant.

The study described in the chapters that follow is intended overcome these limitations.

Since the MusiFind project is the direct progenitor of this thesis it is very important to iterate how its two operating paradigms shaped its research programme. In fact, the operating paradigms of the MusiFind project continue to be those of this thesis. Thus, the implications for research concerning Paradigm 1 (i.e., adoption of the FBIR model) were, and continue to be:

1) the representation of music information in a form amenable for use within traditional FBIR systems (i.e., alpha-numerically in the form of words);
2) the n-gramming of melodic strings into “musical words” (to realize the above);
3) the use and evaluation of search methods traditionally associated with FBIR systems (i.e., inverted file as opposed to linear scanning); and,
4) the evaluation of the system using traditional IR evaluations metrics.

Similarly, Paradigm 2 (i.e., the principle of parsimony) implied, and continues to imply:

1) the preference for interval-only representations (i.e., a simpler solution; a prior solution);
2) the continued preference for the FBIR model (i.e., nothing in the research indicated that it should be abandoned for a more optimal solution); and,

3) the continued preference for the interval-only, n-gram representations (i.e., nothing in the research indicated that it should be abandoned for a more optimal solution).

4.5 Summary

In an effort to better contextualize the research reported upon in the chapters that follow, we have reviewed our extensive prior research into MIR system design and evaluation. We have explicated how the two operating paradigms under which the MusiFind project was conducted shaped all design and evaluation decisions made by the author and his associates. We have highlighted the research methods used by, and the preliminary results derived from, the MusiFind project. We have shown how the findings of the MusiFind project suggested that a more extensive, and more rigorous, evaluation of the MusiFind approach was warranted. We have emphasized the continued importance of the MusiFind project's two operating paradigms on the research conducted for this thesis.

5 Research Questions and Study Overview

5.1 Introduction

This study was built upon and extends the findings of the MusiFind project. The research was conducted in two, chronologically ordered, phases. Phase I was a set of informetric analyses performed upon the interval-only melodic strings found in, and the n-gram databases created from, a sizable collection of folksongs. Phase II was a comprehensive set of experimental information retrieval simulations and evaluations. The Phase II experiments were designed to examine how the n-gramming of interval-only melodic information into "musical words" (i.e., the MusiFind approach) affects the retrieval of the folksongs using a ranking retrieval system.

Notwithstanding that Phases I and II differed significantly in the research methods employed, they were not wholly independent of one another. For example, information from Phase I was used to determine the configurations of intervals used in the various classification schemes that underwent Phase II experimentation. Justification for the choice of the particular term-weighting scheme chosen for use in the retrieval experiments was dependent on the Phase I data. Selection of the n-gram databases that went on to evaluation in Phase II was predicated on the data collected and analyzed in Phase I. Analysis of Phase II results was conducted with reference to the data collected in Phase I.

In this overview chapter we will highlight the salient features of this study, beginning with our principal and subsidiary hypotheses, which are found below. A summary of the special nomenclature used throughout the remainder of this thesis is found in Table 5-1. We will provide a very broad overview of the informetric analyses (Phase I) and the IR simulations and evaluations (Phase II). The details concerning the specific objectives, methods, data, and conclusions pertaining to each phase can be found in Chapter 6 (informetric analyses) and Chapter 7 (IR simulations and evaluations). We will also bring to the fore the major differences between this study and the research performed earlier under the rubric of the MusiFind project (Chapter 4).

Table 5-1. Study nomenclature (Independent Variables)

Factor Name	Short Form	Definition	Code Used	Comments
Classification	CLASS	The number of interval classes used to represent melodies (i.e., size of alphabet)	*C3*	3 interval classes used
			C7	7 interval classes used
			C15	15 interval classes used
			CU	Intervals taken as given in melody
N-gram ***L***ength	NLEN	Number of contiguous intervals in each n-gram	*L4*	length-4 string
			L5	length-5 string
			L6	length-6 string
Query Length	QLEN	Number of contiguous intervals in a string used as a query	*Q6*	length-6 string
			Q8	length-8 string
			Q10	length-10 string
Query Location	QLOC	Position in a song from which a query string is extracted	*I*	***I****ncipit*: query string extracted from a song starting at song's first interval
			R	***R****andom*: query string extracted from a song starting anywhere but song's first interval
Query Quality	QQUAL	Indicates whether query string represents a perfectly formed query or one with an error present	*P*	***P****erfect*: query string taken as extracted from song then subjected to Classification
			E	***Error:*** query string modified with one interval randomly changed prior to Classification

5.2 Principal hypothesis

It was hypothesized that, for purposes of information retrieval, there is enough information contained within the interval-only representation of monophonic melodies that the n-gramming of interval-only melodic strings into "musical words" and their subsequent indexing will allow users the same access to melodic information that indexes of "real words" give to textual information. Both Phases I and II examined this hypothesis.

5.2.1 Subsidiary hypothesis 1

It was hypothesized that there is some type of equivalency between interval-only melodic n-grams (i.e., "musical words") and "real words," intervals and letters. Phase I focused upon this hypothesis.

5.2.2 Subsidiary hypothesis 2

It was hypothesized that the use of some type of ranked retrieval method would overcome any loss of retrieval effectiveness associated with the absence of within-song location information, making it unnecessary to include such information within the indexes. Phase II evaluated this hypothesis.

5.2.3 Subsidiary hypothesis 3

It was hypothesized that application of the classification function *C* would offer a level of "forgiveness" (i.e., resilience to query errors) inversely proportional to the number of classes used to classify the intervals. Phase II evaluated this hypothesis.

5.3 Principal Components

5.3.1 Music database

The database used in this study was the collection of 9354 folksongs created by McNab (McNab et al. 1996). The McNab collection was chosen for three reasons. First, it contains a mix of folksongs that represent American, European, and Asian (i.e., Chinese) traditions. Given this wide range of musical traditions represented, the results should be amenable to generalization, at least within the domain of vocal music. Second, the McNab collection has had duplicate songs removed. Third, the McNab researchers have been kind enough to make their collection available for experimentation and evaluation. In the spirit of Cranfield-style IR evaluations, we hoped that the use of the McNab collection would help establish it as a candidate for use as a standard test collection. Thus established, meaningful comparisons of retrieval methods could be made between the MusiFind approach and any future methods proposed by other MIR researchers.

5.3.2 Retrieval software

The SMART Information Retrieval System Version 11.0 (1992) was selected as the retrieval software. SMART is actually a collection of information retrieval tools that

is very flexible in its configuration (Salton and McGill 1983). This flexibility includes the ability to select various term-weighting schemes for use in ranked retrieval. The ability to handle both batch query processing and relevancy evaluations automatically also recommended the SMART system. The use of the SMART system is also consistent with Paradigm 1 in that it is an "off-the-shelf" system. Finally, the source code to the SMART system is in the public domain and the SMART developers have given experimenters permission to modify the code to suit their particular needs.

5.3.3 Phase I: Informetric analyses [31]

Because the McNab collection comprises melodic representations that include both pitch and rhythm information, reducing the folksongs to their interval-only representation was the first task. After the songs in the McNab database were reduced to their interval-only representations, the second task was the creation of the baseline database (BD) file where each song is represented by a contiguous string of unclassified intervals. The occurrences of each interval type within the BD file was counted.[32] This informetric information was used to assist the author in constructing the classification schemes. Once constructed, the *C3*, *C7*, and *C15* representations of the songs were created. These three representations were then n-grammed, along with the unclassified representation *CU*, yielding the n-gram databases in Table 5-2.

Table 5-2. N-gram databases

C3L4	C7L4	C15L4	CUL4
C3L5	C7L5	C15L5	CUL5
C3L6	C7L6	C15L6	CUL6

[31] Informetric analyses on smaller n-grammed databases led us to conclude that extensive informetric analyses of the McNab et al. collection was warranted (Downie 1997)

[32] When performing informetric analyses, it is customary to make the distinction between *types* and *tokens*. *Types* are the unique entities (e.g., words, symbols, n-grams, etc.) that make up a corpus of interest. *Tokens* are the instances of the types. For example, the collection [dog, dog, cat] contains two types (dog and cat) and three tokens. In this study the types of interest were the distinct n-grams (or intervals) and the tokens the instances of same.

To ascertain in what ways intervals and letters, n-grams and words, might –or might not– be similar, and to predict retrieval performance, descriptive data was collected, and informetric analyses performed upon that data, in an effort to answer the following questions:

1) How many types (and tokens) are present in the databases (i.e., the size of the "alphabets" in the case of intervals; size of "vocabulary" in the case of n-grams)?
2) How are the tokens distributed over the types?
3) How are the types (and tokens) distributed over the songs?
4) How much information is contained, on average, in the types?
5) How specific are the types in denoting the songs in which they are found?
6) How do the databases compare with regard to the ability to discriminate between songs?
7) What effect do the classification and the n-gramming processes have on the answers to the questions above?

5.3.4 Phase II: Information retrieval evaluation

The following factors were examined:

1) Classification (CLASS: *C7, C15, CU*);
2) N-gram Length (NLEN: *L4, L5, L6*);
3) Query Length (QLEN: *Q6, Q8, Q10*);
4) Query Location (QLOC: *I, R*); and,
5) Query Quality (QQUAL: *E, P*).

Stated informally, the preceding factors were examined in an attempt to answer the following questions, respectively:

1) (CLASS) Does the size of the classificatory set used in the creation of the n-gram representations affect performance? If it does not, then one would prefer to represent the melodies using the smallest classificatory set as this would reduce index size. A reduction in index size is to be preferred *ceteris*

paribus. Furthermore, does the size of the classificatory set influence the level of the previously conjectured error "forgiveness"? (See QQUAL).

2) (NLEN) Does the length of the n-gram representations affect performance? If it does not, then one would prefer to represent the melodies using the shortest n-gram length as this would reduce index size. Again, a reduction in index size is to be preferred *ceteris paribus*.

3) (QLEN) How much of the melody will the users have to remember? The probability of query error increases with query length. A MIR system that minimizes the necessary query length, while still performing adequately, is to be preferred *ceteris paribus.*

4) (QLOC) Does the location of the query affect retrieval effectiveness? Can users submit queries that represent internal phrases or must they try to match only the beginnings (i.e., *incipits*) of the melodies? The ability to search for internal phrases would greatly enhance the utility of a MIR system.

5) (QQUAL) Do minor query errors affect performance? A MIR system that minimizes the effect of minor query errors, while still performing adequately, is to be preferred *ceteris paribus.*

5.4 Important changes

There were several important differences between the MusiFind research outlined in Chapter 4 and the present study. The more significant changes are discussed below. Other, primarily minor, changes are noted *ad passim* in Chapters 6 and 7.

5.4.1 Selection of classification schemes

This study investigated different classification schemes than those evaluated in Downie (1995, 1996b). The *C7* and *C12* classification schemes used in the previous studies did not preserve directional information (i.e., whether the intervals went up or down). Dowling (1978) has shown that one of the strongest factors in melodic memory and recognition is the shape, or contour, of a tune. That is to say, users might not remember the specific pitches of a given song but they do have a strong sense of the

direction of its intervals. For this reason, all the classification schemes created and evaluated in this study preserved directional information.

For reasons of statistical validity, the number of elements in each classification scheme should be as distinct as possible.[33] An important factor under investigation in this study is the effect of the size of the classificatory set on retrieval effectiveness (i.e., *Cx*, where *x* is the number of elements in the set). The *C3, C7, C15,* and *CU* set sizes of this study were chosen because they are more clearly differentiated from each other than the *C7, C12, C14,* and *C23* of the earlier studies.

5.4.2 Selection of retrieval method

Instead of the Boolean retrieval operations used in the earlier studies, one of the myriad ranking retrieval methods available in the SMART system was selected for use in the Phase II retrieval tests. According to Harman (1992), ranking retrieval methods have two important features that recommended them over Boolean approaches:

1) adjacency operations or field restrictions, necessary in Boolean systems, are not necessary in ranking systems; and,

2) stoplists are not required, nor recommended, for ranking systems.

Feature 1 offered the potential advantage of not having to store n-gram position locations in the indexes. This afforded a much smaller index than that created for "Boolean systems (in the order of 10% to 15% of the text size)" (Harman 1992). The ability to create smaller indexes can offer speedier searching and is consistent with the principle of parsimony (Paradigm 2) underpinning this study.

---

[33] With any classification scheme the amount of information lost through the collapsing of data into categories is inversely proportional to the number of categories available. It is possible to envision the *C3, C7, C15,* and *CU* sets as lying more evenly spaced upon an information-loss continuum than the *C7, C12, C14,* and *C23* of the earlier studies, with *C3* representing "extreme" information loss, *C7* "high" information loss, *C15* "moderate" information loss, and *CU* "low" information loss. Thus, a test for effects of the new classification CLASS factor becomes, more clearly, a test for the effect of information loss on retrieval effectiveness

Feature 2 is particularly pertinent to music information retrieval. In text, words such as "the," "of," "and," "to," etc., are traditionally included in a stoplist. The ORBIT Search Service has eight [sic] stopwords: "'and,' 'an,' 'by,' 'from,' 'of,' 'the,' and 'with,'" (Fox 1992). Besides being ubiquitous, such stoplist terms are:

1) low in semantic content; and,

2) lacking in discriminatory value.

Given these attributes the exclusion of stoplist terms from an index is generally considered advantageous (or minimally detrimental).

Through the use of informetric analyses it is very easy to isolate the set of the most frequently occurring melodic n-grams. These could be designated as stopwords and then excluded from the indexing process. While this approach would be statistically equivalent to the creation of a text stoplist, the notion of "semantic content," as applied to music n-grams, is more problematic. Simply put, what does a given n-gram ***mean***? Since no satisfactory answer exists to this question, the author thought it prudent to beg the question by foregoing the application of stoplists. Fortunately, ranking systems allow researchers this luxury.

5.4.3 Selection of IR evaluation metrics (Dependent Variables)

The distinction between ***unordered sets*** and ***ordered lists*** was the key factor in selecting the normalized precision (NPREC) and normalized recall (NREC) metrics over the standard precision and recall measures. Unlike Boolean systems, which in response to a given query, present to the user an ***unordered subset*** of a *N*-document database in the form of the set of retrieved documents, ranking systems present to the user an ***ordered list*** of all the documents in a *N*-document database, ordered from 1 to *N*, with the first-ranked item representing the document that the system has determined as having the strongest similarity with the query.[34] The ideal quality in a ranking system is the ability to place in the top portion of its ordered list only, and all, members of the set of *m* relevant

[34] This is true in theory, at least. In general practice, however, ranking systems usually limit the length of the returned listing using some predetermined threshold of similarity between query and documents.

documents available in its database (i.e., the set of items listed at ranks 1 through *m* contains only relevant documents).

The standard precision and recall metrics are measures involving ratios of the ***unordered sets*** of retrieved, relevant, not-retrieved and not-relevant documents making them well-suited for evaluating Boolean systems. Using arbitrary cut-off points (e.g., the first 10, 20, or 30 documents listed, etc.), it is possible to apply standard precision and recall metrics in the evaluation of ranking systems. However, standard precision and recall leave unevaluated the central feature of interest in a ranking retrieval system, namely, the quality of the document rankings created by its ranking algorithm (i.e., how close a system's ***ordered list*** is to the ideal ***ordered list***).

The normalized precision (NPREC) and normalized recall (NREC) metrics were created specifically to capture how closely a ranking system performs relative to the ideal by including in their calculation information about the ranks at which relevant documents are listed. A NPREC or NREC value of 1 indicates that the ideal has been realized while a value of 0 indicates the worst case (i.e., all the relevant documents are assigned to the very bottom of the ordered list). The NPREC and NREC metrics are defined as:

$$\mathrm{NPREC} = \mathrm{PRECISION}_{\mathrm{norm}} = 1 - \frac{\sum_{m=1}^{\mathrm{REL}} \log \mathrm{Rank}_m - \sum_{m=1}^{\mathrm{REL}} \log m}{\log \mathrm{N!}/(\mathrm{N}-\mathrm{REL})!\mathrm{REL}!}\text{; and,}$$

$$\mathrm{NREC} = \mathrm{RECALL}_{\mathrm{norm}} = 1 - \frac{\sum_{m=1}^{\mathrm{REL}} \mathrm{RANK}_m - \sum_{m=1}^{\mathrm{REL}} m}{\mathrm{REL}(\mathrm{N}-\mathrm{REL})}$$

where N is the number of documents in the database, REL the number of relevant documents contained in the database, and RANK_m the rank assigned to relevant document *m.*[35]

[35] Equations adapted from Salton and McGill (1983).

5.4.4 Introduction of sensitivity evaluation (QQUAL)

McNab et al. (1996) noted that, during the course of their investigations, many of their experimental users introduced a variety of errors into their sung queries. Query errors are a fact in any IR environment; however, in the context of MIR they are particularly problematic. It is reasonable to assume that users of text retrieval systems have much more expertise in matters of language than users of *Locating* MIR systems have in matters of music. If an error occurs in a text query (e.g., a misspelling) users generally have the linguistic wherewithal to correct the error. Music queries, especially those that are sung by non-musicians with poor notational skills, are less amenable to correction.

In order to make the Phase II IR simulation and evaluation experiments more representative of real-world conditions, a new experimental factor (QQUAL) was introduced. The QQUAL factor was created to evaluate the "brittleness" or "sensitivity to errors" of the various configurations of databases and queries under investigation. "Fault tolerance" (i.e., "forgiveness") is another phrase commonly applied to such evaluations. A system that "breaks down" (i.e., fails to function properly) in the presence of minor errors is said to be "brittle" or "sensitive to errors". A "fault tolerant" system, on the other hand, is one that continues to function within specified performance limits despite minor errors. In mission-critical applications, such as air traffic control or certain health information systems, "brittleness" can literally be a matter of life or death. Within the domain of MIR, while not a matter of life or death, it is still important to be cognizant of potential areas of brittleness. Design recommendations must be made with knowledge of system sensitivities if MIR systems are to be created that perform as well in the real world as they do in the research lab.

We heuristically classified the types of errors noted by McNab et al. (1996) as *Expansion*, *Compression*, *Repetition* and *Omission* errors. These error classes became the basis for the simulation of user errors under the *Error* condition (*E*) of the QQUAL factor. Below are the characteristics associated with each error class and the method we used to simulate errors of each type. The detailed algorithmic summary of the error simulation process is found in Chapter 7.

1) *Expansion* errors occurred when the subjects sang the smaller intervals that fell within the 1 <= *n* <= 4 (Small Positive) and -1 >= *n* => -4 (Small Negative) ranges. Subjects tended to expand these intervals slightly. For example, a minor third (*n* = ±3) might be sung as a major third (*n* = ±4). Expansion errors were simulated by replacing n with *n*+1 in the positive case and *n*-1 in the negative case.

2) *Compression* errors occurred when the subjects sang the larger intervals that fell within the *n* >= 5 (Large Positive) and *n* <= -5 (Large Negative) ranges. Subjects tended to compress these intervals slightly. For example, a minor sixth (*n* = ±8) might be sung as a perfect fifth (*n* = ±7). Compression errors were simulated by replacing *n* with *n*-1 in the positive case and *n*+1 in the negative case.

3) *Repetition* errors occurred when subjects incorrectly repeated notes (*n* = 0). In general, users inserted repetitions based upon their misperceptions of a song's syllables. Repetition errors were simulated by replacing *n* with *n* = 0.

4) *Omission* errors (the most infrequent error type) occurred when users simply omitted an interval. For example, about half of the McNab subjects omitted a descending interval from the song "Yankee Doodle". Omission errors were simulated by deleting *n* from the query string.

5.5 Summary

In this overview chapter we have outlined the salient features of the thesis research proper. The McNab folksong collection and the SMART retrieval system were brought together to evaluate our principal hypothesis that the n-gramming of interval-only melodic strings into "musical words" is a feasible MIR solution. Five experimental factors —CLASS, NLEN, QLEN, QLOC, and QQUAL— were constructed to evaluate our principal hypothesis, the series of subsidiary hypotheses implied by the principal hypothesis, and the informal questions arising from these hypotheses.

Informetric analyses (Phase I) were undertaken to:

1) assist in the creation of the classification schemes;

2) provide data for examining the hypothesis that “musical words” and “real words” have some type of equivalency (Subsidiary hypothesis 1); and,

3) provide data for explicating the results in Phase II.

IR simulations and evaluations (Phase II) were undertaken to:

1) examine whether there is enough information contained within the interval-only representation of monophonic melodies that the n-gramming of interval-only melodic strings into “musical words” and their subsequent indexing (i.e., the MusiFind approach) offers the same access to music information that indexes of “real words” give to textual information (Principal hypothesis);

2) examine whether the use of a ranking retrieval system overcomes any loss of retrieval effectiveness associated with the absence of within-song location information (Subsidiary hypothesis 2): and,

3) examine the ability of the MusiFind approach to withstand minor query errors (Subsidiary hypothesis 3).

We have also noted the important differences, and the reasons for those differences, between the present research and that conducted earlier under the rubric of the MusiFind project. These major modifications included:

1) new classification schemes that were more distinct and incorporated directional information;

2) the use of a ranking retrieval system instead of a Boolean system;

3) the use of the NPREC and NREC evaluation metrics in place of the standard precision and recall measures; and,

4) the formal examination of the ability of the MusiFind approach to withstand minor query errors (i.e., sensitivity evaluation).

6 Phase I: Informetric Analyses

6.1 Introduction[36]

The best justification for performing informetric analyses in conjunction with the simulation and evaluation of IR systems can be found in Wolfram's *Applying informetric characteristic of databases to IR system file design, Parts I and II* (1992a, 1992b). Knowledge of the informetric properties of an IR system has been shown to assist researchers and developers predict and model:

1) storage requirements (Tague 1988; Tague and Nicholls 1987);

2) index growth (Heaps 1978; Wolfram et al. 1990);

3) optimal indexing and search strategies (Sampson and Bendell 1985; Wolfram 1992b); and,

4) query usage (Nelson 1988; Wolfram 1992b).

Of these four, the first three were seen by us as being germane to our needs, and were therefore influential in the structuring of this study. The third item, in particular, was the impetus behind our own analyses. The fourth, while important, was not part of the present set of informetric analyses; however, it does remain on our future research agenda.

In this chapter, we will begin with the details concerning the wide range of methods employed to conduct our informetric analyses of the baseline database and the various n-gram databases created from it. We will then present the data and highlight their significant features. We will discuss how the data influence our thinking concerning the hypothesized equivalencies between letters and intervals, words and n-grams. We will also discuss how the findings inform us about the potential strengths and weaknesses of the MusiFind approach to MIR. We will conclude this phase of the study with an explication of how the data collected in these informetric analyses influenced the Phase II retrieval evaluations.

---

[36] This chapter is a comprehensive revision of Downie (1998).

6.2 Methods

6.2.1 Analytic tools

The primary analytic tools were a series of PERL programmes written by the author and his assistant, Kevin Kennedy (e.g., descriptive statistics, entropy measurements, etc.). The term discrimination analyses were conducted using a PERL programme written by Dr. David Dubin (Dubin 1997). Distribution modeling was performed using an adaptation of a model-fitting programme written and adapted by Dr. Michael Nelson.

6.2.2 Descriptive statistics on intervals (type = interval)

The first types of interest were the intervals found in the Baseline Database (BD) file. Data were gathered concerning:

1) the number of types present; and,
2) the distribution of tokens over types.

Because these data were used to assist in the creation of the classification schemes, we present them now in Figure 6-1, Table 6-1, and Table 6-2.

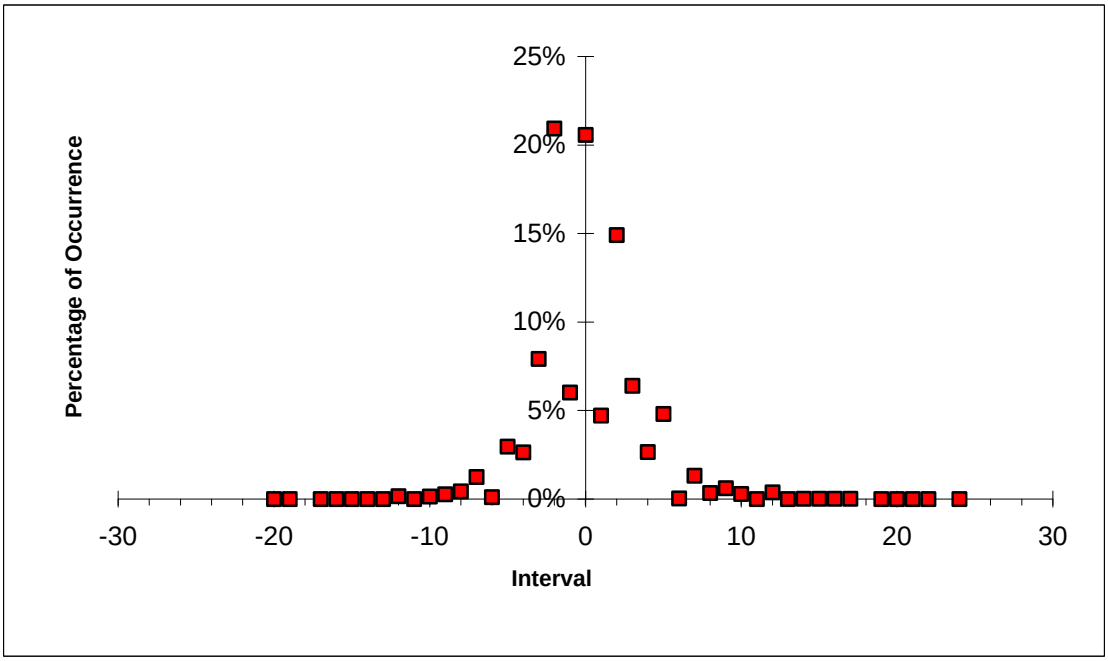


Figure 6-1. Distribution of interval tokens over interval types

Table 6-1. Descriptive data concerning intervals

	Interval Data (per song)		Unique Intervals (per song)
Mean	53.78		10.50
SD	29.37		2.27
Range	494		21
Minimum	7		3
Maximum	501		24
Total	503,086		

Table 6-2. Summary of interval occurrence and classification schemes

Negative Interval Classification Codes						Positive Interval Classification Codes					
C3	**C7**	**C15**	**CU**	**-Interval**	**(Probability)**	**+Interval**	**(Probability)**	**CU**	**C15**	**C7**	**C3**
a	a	a	a	0	(0.2056)	0	(0.2056)	a	a	a	a
b	c	b	b	-1	(0.0603)	1	(0.0473)	B	B	C	B
b	b	c	c	-2	(0.2093)	2	(0.1491)	C	C	B	B
b	c	d	d	-3	(0.0792)	3	(0.0640)	D	D	C	B
b	d	e	e	-4	(0.0264)	4	(0.0265)	E	E	D	B
b	d	f	f	-5	(0.0297)	5	(0.0480)	F	F	D	B
b	d	g	g	-6	(0.0010)	6	(0.0004)	G	G	D	B
b	d	h	h	-7	(0.0125)	7	(0.0132)	H	H	D	B
b	d	h	i	-8	(0.0043)	8	(0.0034)	I	H	D	B
b	d	h	j	-9	(0.0028)	9	(0.0061)	J	H	D	B
b	d	h	k	-10	(0.0015)	10	(0.0030)	K	H	D	B
b	d	h	l	-11	(0.0000)	11	(0.0001)	L	H	D	B
b	d	h	m	-12	(0.0017)	12	(0.0037)	M	H	D	B
b	d	h	n	-13	(1.19E-05)	13	(0.0000)	N	H	D	B
b	d	h	o	-14	(0.0001)	14	(0.0002)	O	H	D	B
b	d	h	p	-15	(3.58E-05)	15	(0.0001)	P	H	D	B
b	d	h	q	-16	(2.39E-05)	16	(0.0001)	Q	H	D	B
b	d	h	r	-17	(2.58E-05)	17	(0.0001)	R	H	D	B
b	d	h	t	-19	(3.98E-06)	19	(3.18E-05)	T	H	D	B
b	d	h	v	-20	(7.95E-06)	20	(5.96E-06)	U	H	D	B
						21	(3.98E-06)	V	H	D	B
						22	(3.98E-06)	W	H	D	B
						24	(3.98E-06)	Y	H	D	B

6.2.3 Creation of classification schemes

The *CU* classification scheme was created using the intervals as they occur in the BD file. However, the knowledge gained through the examination of the intervals in the BD file became crucial in the creation of the remaining classification schemes. Because

there exists a substantial group of interval types that occur very infrequently, it became necessary to develop a set of heuristics to apply such that the resulting classification schemes pooled these rare events into classes that most efficiently represented the melodies, given the prior constraint that directional information be preserved. Efficiency is defined here as a classification scheme's ability to minimize the amount of information lost through its classification process.

An important metric of efficiency is Shannon's measure of a system's self-information, or *entropy* (Shannon and Weaver 1949):

$$\bar{H} = -\sum_{r=1}^{n} p_r \log_2 p_r$$

where $\bar{H}$ is the entropy of a collection of n types (or classes) r, and p_r is the probability of occurrence of type (or class) r. The unit of measurement for entropy is the *bit*.[37]

A system (e.g., a classification scheme) is most efficient when $\bar{H}$ is maximized. $\bar{H}$ is maximized when the probabilities of occurrence for each type (or class) r are equal. When situations arise where the probabilities cannot be equalized (e.g., given the presence of a prior constraint), $\bar{H}$ is maximized when the differences between each p_r are minimized.

Thus, the remaining classification schemes created (i.e., *C3*, *C7*, and *C15*) were those that most efficiently represented the melodies because —within the prior constraint that directionality be preserved— the intervals were grouped such that $\bar{H}$ was maximized for each scheme.

6.2.4 Entropy of intervals in BD file

As stated before, entropy is maximized when the probabilities of occurrence of each type are equal. In such a case the entropy of a system can be expressed as:

---

[37] Speaking of a system's entropy as being x bits is another way of saying that, on average, a type within the system contains x bits of information.

$$F_0 = -\log_2(1/n)$$

where F_0 is the entropy of a collection of equally distributed n types. F_0 is also used to calculate the entropy of a system where the number of types is known but the distribution of tokens is not. When the probabilities of occurrence are known for the n types in a collection the $\bar{H}$ equation is used. The symbol F_1 is sometimes used in place of $\bar{H}$ to signify that observed probabilities were used in the calculations (under the assumption of type independence).[38]

Assuming an English alphabet of 26 letters (case excluded), the F_0 of a letter is 4.70 bits. Using observed letter occurrence probabilities (i.e., 1-gram probabilities), Shannon (1951) calculated the entropy of a letter in English text as F_1 = 4.14 bits. Similar F_1 values have been estimated for "other Roman alphabet languages" (Losee 1990). While there are methods other than the simple F_1 equation to estimate entropy,[39] Shannon's F_1 equation, and his resultant value of 4.14 bits, were chosen as the basis for comparing letter and interval entropies because of the equation's conceptual simplicity and the general acceptance of 4.14 bits as a valid entropy estimate.

6.2.5 Entropy of intervals in n-grammed representations

Once the n-grammed databases were created, a second set of data concerning the distribution of interval tokens over types was collected. Ordinarily it would be possible

---

[38] More specifically, F_n signifies that information about the order-n type probabilities (i.e., type dependencies) was used when calculating system entropy. Thus, F_1 denotes entropy calculated using the 1-gram probabilities; F_2 using 2-gram data (i.e., the probability of a type given the probability of the preceding type); F_3 using 3-gram data (i.e., the probability of a type given the probability of the preceding 2-gram); etc., *ad infinitum*. Because the result of any entropy calculation is dependent upon the amount of probability information available, any F_n value calculated is actually only an approximation, or estimate, of a system's true entropy. Only F yields a system's true entropy because it is based upon *all possible* probability information. For the purposes of this thesis, comparing text and intervals up to the F_1 level was sufficient.

[39] The different methods each yield different entropy estimates (see Heaps 1978; Losee 1990). Notwithstanding that different entropy estimates exist, as long as the same method is used in each case (e.g., application of the F_1 equation), valid comparisons can be made.

to derive mathematically the descriptive statistics for each of the classification schemes, given the data collected from the BD file, since the mapping of a classification scheme to raw intervals is known. However, n-gramming introduces certain distortions that must be taken into account. Consider the following contiguous melodic string and its n-gram representations where I_x represents any unique interval at position *x* in the string:

Contiguous string:

$I_1\ I_2\ I_3\ I_4\ I_5\ I_6\ I_7\ I_8\ I_9\ I_{10}\ I_{11}$

4-gram representation:

$[I_1\ I_2\ I_3\ I_4]\ [I_2\ I_3\ I_4\ I_5]\ [I_3\ I_4\ I_5\ I_6]\ [I_4\ I_5\ I_6\ I_7]\ [I_5\ I_6\ I_7\ I_8]\ [I_6\ I_7\ I_8\ I_9]\ [I_7\ I_8\ I_9\ I_{10}]\ [I_8\ I_9\ I_{10}\ I_{11}]$

5-gram representation:

$[I_1\ I_2\ I_3\ I_4\ I_5]\ [I_2\ I_3\ I_4\ I_5\ I_6]\ [I_3\ I_4\ I_5\ I_6\ I_7]\ [I_4\ I_5\ I_6\ I_7\ I_8]\ [I_5\ I_6\ I_7\ I_8\ I_9]\ [I_6\ I_7\ I_8\ I_9\ I_{10}]\ [I_7\ I_8\ I_9\ I_{10}\ I_{11}]$

6-gram representation:

$[I_1\ I_2\ I_3\ I_4\ I_5\ I_6]\ [I_2\ I_3\ I_4\ I_5\ I_6\ I_7]\ [I_3\ I_4\ I_5\ I_6\ I_7\ I_8]\ [I_4\ I_5\ I_6\ I_7\ I_8\ I_9]\ [I_5\ I_6\ I_7\ I_8\ I_9\ I_{10}]\ [I_6\ I_7\ I_8\ I_9\ I_{10}\ I_{11}]$

Figure 6-2. N-grams of intervallic information

Note here that the frequency of each interval I_x in the *contiguous string* is equal to 1. This is not the case for the interval frequencies in the n-grammed representations because a pattern of frequency distortions is introduced. For example, in the 5-gram case I_1 occurs in one n-gram, I_2 in two n-grams, I_3 in three n-grams, etc. These distortions are summarized in Table 6-3.

Table 6-3. Frequency distortion factors caused by n-gramming

Interval Position (11 interval song)	Frequency distortion factor		
	4-gram	5-gram	6-gram
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	4	5	5
6	4	5	6
7	4	5	5
8	4	4	4
9	3	3	3
10	2	2	2
11	1	1	1

As one can see in Table 6-3, the distortions are not uniform. The distortions caused by n-gramming are characterized by a relative under-representation of the

intervals that occur close to the terminal positions of the contiguous string. The regions bounded by *L* - 1 positions (where *L* is the length of the n-gram) from either end of the contiguous string contain the under-represented intervals. All other interval counts, while inflated from the contiguous case, are not distorted by n-gramming.

Had the frequency distortions been uniform no special analytical problem would have existed. If the relative frequencies of the intervals had remained constant, then their probabilities of occurrence would not have been affected. However, as Table 6-3 clearly shows, this was not the case; therefore, an empirical examination of the interval entropies *as they are found in the n-grams* was warranted. Also, by comparing the entropy value calculated for the intervals in the BD file with the values calculated for the *CUL4*, *CUL5*, and *CUL6* databases, we can determine the amount of information loss, or gain, that is directly attributable to the n-gramming process.[40]

6.2.6 Descriptive statistics on n-grams (type = n-gram)

N-grams taken from each of the n-grammed databases were the second types of interest. Data were collected concerning:

1) the number of types present in each database;

2) the number of tokens present in each database;

3) distribution of tokens over types in each database;

4) the number of types present in each song;

5) the number of tokens present in each song; and,

6) the distribution of types over songs.[41]

---

[40] This is the case because the *CU* databases did not undergo any classification processing, so any effects noted would be caused solely by n-gramming.

[41] Only one token for each type present in a song was included in this calculation, thus removing the effect of multiple tokens representing the same type (i.e., within-song redundancy). This was done to better understand how the types might be used as individual signifiers of the songs in which they are found (i.e., the specificity of the n-grams types).

The data collected can be divide into two *views*: View A (Items 1, 2 and 3) and View B (Items 4, 5, and 6). View A data were collected to learn more about the general nature of the databases as representative collections of folksongs, much in the same way that one performs *corpus* linguistic studies to learn more about the nature of a body of textual works or a language (e.g., Zipf (1949) or Dewey (1950)). View B data were collected to evaluate the characteristics of the individual songs with an eye toward determining the efficiency of the n-grams as signifiers of individual songs (i.e., to explore for characteristics that might influence the indexing and retrieval of the songs).

6.2.7 Entropy of n-grams

The View A calculation of the entropy of each n-gram database was straightforward. The probability of occurrence of an n-gram type was simply its frequency of occurrence divided by the total number of tokens in its database. The probabilities were thus calculated for each n-gram type in each n-gram database. The entropy of each database was then calculated using these probabilities and the $\overline{H}$ equation.

The View B calculation of entropy was slightly different from the View A calculation. The probability of occurrence of an n-gram type was calculated using the number of songs in which the type occurred divided by the number of tokens in the database where each song was represented by only one token of any given type (i.e., the total number of tokens in the database once all n-gram duplicates were removed from each song). The View B entropy for each n-gram database was then calculated using these adjusted probabilities and the $\overline{H}$ equation.

Shannon (1951) used the famous Zipf (1949) equation to estimate the entropy of an English word as 11.82 bits, assuming a vocabulary size of 8,727 words.[42] Yavuz (1974), however, using a more sophisticated approach to estimating the appropriate

---

[42] The Zipf equation can be found in Chapter 6.2.9. Zipf's findings suggest that the probability P_r of the r^{th} word in a ranked list of terms (descending from the most frequent) is approximately $0.1/r$. In order to best estimate the entropy of words it is necessary to select a vocabulary size N such that $\sum_{r=1}^{N} 0.1/r$ is as close to unity as possible.

vocabulary size *N*, arrived at *N* =12,366 words to estimate word entropy as 9.72 bits. Since Gringnetti (1964) assumed 12,370 words to also estimate word entropy as 9.72 bits, we decided to take this value (9.72 bits) as the base entropy estimate against which the n-gram data would be compared.

6.2.8 Term discrimination analysis

Srinivasan (1992) defines the Discrimination Value (*DV*) of a term as:

> ...the degree to which a term is able to discriminate or distinguish between the documents of the collection as described by Salton and Yang (1973). The more discriminating a term, the higher its value as an index term. The overall procedure is to compute the average interdocument similarity in the collection, using some appropriate similarity function. Next, the term *k* being evaluated is removed from the indexing vocabulary and the same average similarity is recomputed. The discrimination value (DV_k) for the term is then computed as:
>
> DV_k = (Average similarity without *k*) - (Average similarity with *k*).

This equation can be used to generate two classes of terms:

1) Good discriminators where DV_k is positive; and,

2) Poor discriminators where DV_k is negative.

The percentage of good and bad discrimination values for each n-gram group was determined by running the n-gram databases through Dubin's *TDV.pl* programme (1997). *TDV.pl* uses the *exact centroid method* as outlined in Crouch (1988) to compute the average document vector for the entire collection. Once the average document vector had been computed, the average similarity for a collection with all n-grams present, or *Q*, was then calculated by comparing each song to the average song vector using the cosine correlation metric, below:

$$\mathrm{COSINE}(\mathrm{DOC_i}, \mathrm{DOC_j}) = \frac{\sum_{k=1}^{t} (\mathrm{TERM_{ik}} \bullet \mathrm{TERM_{jk}})}{\sqrt{\sum_{k=1}^{t} (\mathrm{TERM_{ik}})^2 \bullet (\mathrm{TERM_{ik}})^2}}$$

Next, one n-gram *k* was removed. After the removal, the songs were then recompared to the average song vector, which yielded a DV_k for the removed n-gram.

The removed n-gram was replaced, and another selected for removal. This process continued until all n-grams had been analyzed, thus calculating a DV_k for each n-gram.

On a broader scale, the *Q* value calculated for each n-gram database is also informative. According to Salton (1975), a database can be seen as creating a document space where each document represents a point in that space. A collection of documents that are similar to one another creates a space that is said to be dense. As the similarity of the documents decreases, the density of the space decreases. Denser document spaces are considered to be poorer candidates for effective information retrieval (i.e., it is more difficult to properly discriminate between the various documents). Since *Q* is a measure of average document similarity, it is also a measure of a database's density. Thus, knowledge of a database's *Q* value can be used as a predictor of potential retrieval performance. A slight modification was made by us to Dubin's *TDV.pl* so that the *Q* value for each database was explicitly recorded.

6.2.9 Informetric modeling

The distribution of tokens over types where the types were the n-grams found in each of the databases was modeled. The size-frequency approach to distribution modeling (Tague 1990) was used. The method for estimating the parameters was minimum Chi-square obtained by numerical methods.[43] Four distributions were tested for goodness-of-fit. The proportion of terms (i.e., n-grams) *f*(*x*) appearing *x* times were estimated by:

a) Simple Zipf distribution:

$$f(x) = a / x^b, \quad x = 1,2,3....$$

where *a* and *b* are constants determined by the data;

b) Mandelbrot-Zipf distribution (MZ):

$$f(x) = a / (x + c)^b, \quad x = 1,2,3....$$

---

[43] We gratefully acknowledge Dr. Nelson's work in adapting his original model-fitting program to our needs.

which is a generalized three parameter form of the simple Zipf, where *a*, *b*, and *c* are constants determined by the data;

c) Zero-truncated Generalized Waring distribution (GW):

$$f(x) = \frac{\Gamma(x+n)}{B(a,b)\Gamma(n)} \frac{\Gamma(x+n)\Gamma(x+b)}{\Gamma(x+b+n+a)x!}, \quad x = 0,1,2,3,...$$

where *a*, β and ν are constants determined by the data. Because only term frequencies greater than zero were observed, the zero-truncated form of the distribution was obtained by dividing *f*(*x*) by 1-*f*(*0*). The value of *f*(*0*) was easily calculated recursively from *f*(*1*) (see Nelson 1989).

d) Zero-truncated Generalized Inverse Gaussian-Poisson (GIGP) distribution:

$$f^*(x) = \frac{(1-q)^{g/2}}{K_g(a[1-q]^{1/2})} \frac{(aq/2)^x}{x!} K_{x+g}(a) \times \left[1 - \frac{(1-q)^{g/2}}{K_g(a[1-q]^{1/2})} K_g(a)\right]^{-1},$$
$$x = 1,2,3,...$$

where *f**(*x*) indicates the zero truncation (as above), K_γ denotes the modified Bessel function of order γ (see Burrell and Fenton 1993), α, θ, and γ are constants determined by the data.

When performing distribution modeling using the Chi-square goodness-of-fit method, the traditional practice for acceptance of a model distribution is to ***reject*** all those models where the observed C^2 value is greater than, or equal to, the critical C^2 value (at $p = 0.05$ and the appropriate degrees of freedom for the data). Some have argued that this method of determination is too sensitive (i.e., biased toward rejection) because, given even a moderately-sized data set, it is rather easy to exceed the critical C^2 value (Wolfram 1992a; Burrell and Fenton 1993). Notwithstanding this criticism, we decided to follow the traditional practice because of its more conservative approach.

6.3 Observations and analysis

6.3.1 Descriptive interval data and analyses

The mean number of intervals per song, 53.78 tokens/song, along with the mean number of types per song, 10.50 types/song, give us the first indication that there are

some pronounced differences between text and music. First, that a complete "idea" could be represented in such a compact manner is worth noting. For example, *ad extremis*, one song is only 7 intervals long. However, repetition of the complete melody, as one cycles through a series of verses, could exaggerate the significance of this extreme representational compactness. Second, that there are only 10.50 interval types used per song, on average, suggests another dissimilarity between intervals and letters. One would be hard-pressed to imagine a collection of "real" sentences that contained only 11 different letters on average.

The $F_{0(BD)}$ for the 42 intervals found in the baseline database was calculated to be 5.39 bits. This value is greater than Shannon's 4.70 bits because the number of interval types is greater than the 26 alphabetic characters (i.e., letter types) Shannon used in his calculations. However, $F_{1(BD)}$ was calculated for the baseline database as 3.39 bits while Shannon's $F_{1(Shannon)}$ value is greater at 4.14 bits. Thus, intervals in the baseline database contain less information on average than letters do in "real" text, a difference of 0.75 bits (4.14 – 3.39 = 0.75 bits). The greater discrepancy between the intervallic $F_{0(BD)}$ and $F_{1(BD)}$ values (5.39 – 3.39 = 2 bits) than between the Shannon $F_{0(Shannon)}$ and $F_{1(Shannon)}$ values (4.70 – 4.14 = 0.56 bits) can be attributed to the more pronounced skew in the distribution of the interval tokens over types in the BD file than that of the alphabetic tokens found in text. In a rank-probability presentation of the data (Figure 6-3) this skewing is characterized by:

a) the clustering of the majority of interval tokens about the highest ranks (i.e., ranks 1, 2, or 3); [44]

b) a steep downward slope through upper to the lower ranks; and,

c) a long "tail" of very infrequently occurring types.[45]

---

[44] This clustering about the higher ranks creates the prominent "head" of the distribution.

Note in Table 6-2 that the most frequently occurring interval type, "-2", represents nearly 21% of all the tokens in the database. Within the context of music, this finding is as one would expect. According to Western music theory, a "well-constructed" melody is characterized by a series of rapid ascents in pitch followed by longer stepwise descents (i.e., predominantly intervals of "–2", with some "–1" intervals).[46] These descents eventually make their way back to the starting, or tonic, pitch. That the number of positive interval types (22 types) is greater than the number of negative types (19 types), while the proportion of negative tokens (43% of tokens) is greater than the positive tokens (37% of tokens), is also consistent with the theory of well-constructed melodies. That is, one would expect to see a wide variety of individual upward leaps (i.e., relatively fewer tokens representing a wider variety of positive types), each of which should be followed by a group of intervals making a stepwise descent (i.e., more negative tokens representing a smaller variety of negative types), which is precisely what these data indicate. The "0" interval also figures prominently at just under 21% of tokens. The prominence of the "0" interval reflects the very common practice of syllabification of individual lyric words at a fixed pitch. For an example that represents both a) the notion of rapid ascents followed by stepwise descents; and, b) syllabification, sing to yourself the children's classic, *Twinkle, Twinkle, Little Star.*

Dewey (1950) determined that the three most frequently occurring letter types, "E", "T", and "A", represent 12.7%, 9.8% and 7.9% of letter tokens, respectively. In the baseline database, the three most frequently occurring types are "-2", "0", and "+2", and represent 20.9%, 20.6%, and 14.9% of the interval tokens, respectively. That only 3 of the 42 interval types in the BD file account for 56.4% of all the tokens, as compared to

---

[45] A distribution with the above characteristics is also known as a "Zipfian" distribution, after the statistical linguist George Zipf whose analysis of such distributions gave rise to the now-famous Zipf equation (Zipf 1949). We use the term here to denote the shape of the distribution and not to claim that the data can be formally described by the Zipf distribution.

[46] Heinrich Schenker (1868-1935) was the principal exponent of this notion of melodic linearity.

30.4% for text, is another indication that the entropy for intervals must be less than the entropy for letters.

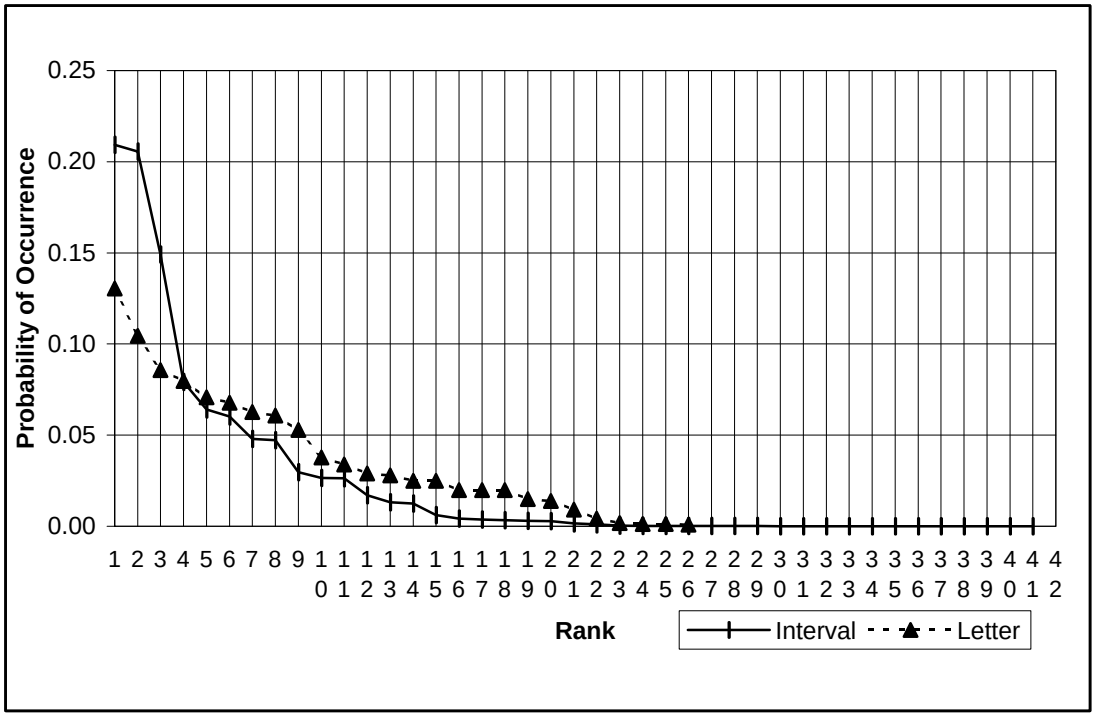


Figure 6-3. Rank-probability comparison of intervals and letters.

Table 6-4. Interval entropy values from n-grammed databases

	F_0	F_1	F_1-F_0	$F_{1(BD)}$-$F_{1(n\text{-grammed})}$	$F_{1(Shannon)}$-$F_{1(n\text{-grammed})}$
C3L4	1.5850	1.5224	-0.0625	-1.8666	2.6176
C3L5	1.5850	1.5225	-0.0624	-1.8666	2.6175
C3L6	1.5850	1.5228	-0.0621	-1.8662	2.6172
C7L4	2.8074	2.7343	-0.0731	-0.6548	1.4057
C7L5	2.8074	2.7346	-0.0728	-0.6545	1.4054
C7L6	2.8074	2.7347	-0.0726	-0.6543	1.4053
C15L4	3.9069	3.2790	-0.6279	-0.1101	0.8610
C15L5	3.9069	3.2795	-0.6274	-0.1096	0.8605
C15L6	3.9069	3.2798	-0.6271	-0.1093	0.8602
CUL4	5.3923	3.3928	-1.9995	0.0037	0.7472
CUL5	5.3923	3.3939	-1.9984	0.0048	0.7461
CUL6	5.3923	3.3948	-1.9975	0.0057	0.7452

The strongest cause of information loss was the application of the classification schemes, with *C3* having an $F_{0(C3)}$ of only 1.58 bits, while the $F_{0(CU)}$ for *CU* is 5.39 bits (Table 6-4). The slight difference in *C3*'s $F_{0(C3)}$ and $F_{1(C3)}$ values (approximately 0.06

bits, rounded) indicates that distribution of negative, neutral, and positive intervals is relatively equal at 43%, 21%, and 37%, respectively. Differences between the F_0 and F_1 values for each scheme increase as *C* increases despite the fact that the classification schemes were created with an eye toward maximizing interval entropy (i.e., shortening the tails, and equalizing the frequencies across the distributions) because, as *C* increases, there are fewer opportunities to collapse rare intervals into larger classes. It is this collapsing that smoothes and trims the distributions.[47] Thus *CU*, where there was no collapsing of interval classes, retained its steep slope and "untrimmed" tail. *CU*'s steep slope and untrimmed tail gives it the largest F_0 to F_1 entropy drop (5.39-3.39 = 2.00 bits, rounded).

The modest difference between the $F_{1(CU)}$ and $F_{1(C15)}$ values (3.39 - 3.28 = 0.11 bits, rounded) is less than we had anticipated. Recall that *CU* has 42 interval types, while *C15* has, of course 15 interval types. It appears that the 27 interval types (42 - 15 = 27) that were folded into the *C15* scheme, but remained unique in the *CU* scheme, occur so infrequently that collapsing of them into larger classes has very little effect on smoothing out the distribution of tokens over types, despite shortening the tail of the distribution.

We assert that this minimal difference between $F_{1(CU)}$ and $F_{1(C15)}$ is an artefact of the style of music represented in the database, that is, vocal music of the folksong genre. Folksongs (for the most part) are compositions that are written by, and performed by, untrained musicians. Since it is difficult for the untrained voice to extend itself over a broad range of intervals, it follows that the range of intervals found in a collection of folksongs would be rather limited. Our data in Table 6-4 is in accordance with this assertion. That is, while there are many different interval types present ($types_{(CU)} = 42$), including some that are very large (e.g., "+24", "-20", etc.), the actual occurrence of the larger and extreme intervals is very rare. Writing as an erstwhile flautist, we make the observation that had this database comprised a collection of flute solos, the occurrence of the larger and extreme intervals would have been much less rare, perhaps even common,

---

[47] Recall that entropy is maximized when the all type frequencies are equal. A distribution of equally frequent types has a slope of $\frac{\Delta y}{\Delta x} = 0$.

because on the flute, as with most instruments, there is no special problem posed by making leaps of arbitrary size. All this being said, we acknowledge the slight difference between the $F_{1(CU)}$ and $F_{1(C15)}$ values but suggest here that the *C15* classification scheme should not be ruled out as a potentially useful scheme based solely on these, genre-specific, data.

As stated earlier, the process of n-gramming under-represents those intervals that occur at the beginnings and endings of songs. The entropy of the intervals was calculated from the intervals counted from the n-grams to ascertain the amount of distortion caused by n-gramming. In Table 6-4, by reading down the column labelled $\mathbf{F_{1(BD)}\text{-}F_{1(n\text{-}grammed)}}$ and noting the differences in values among the *CUL4*, *CUL5* and *CUL6* groups, one can determine the amount of interval information lost (or gained) by n-gramming. The choice of the *CU* allows us to remove the influence of classification on the change in entropy because the *CU* schemes had no classification process applied to the intervals. The amount of distortion caused by n-gramming was extraordinarily slight (i.e., changing at the third decimal place). For these data, it appears that n-gramming causes a drop in entropy of approximately 0.0011 bits for each unit increase in n-gram length. Such a slight distortion need not concern us further.

6.3.2 Descriptive interval data and analyses: concluding remarks

Despite a larger number of interval types present, 42 types (intervals) vs. 26 types (letters), the actual number of interval types used in a given song is markedly lower at 11 types/song, on average, than one would find with letters in a similarly-sized selection of text. Notwithstanding the application of the classification process, which always results in a drop in entropy, intervals contain less information than letters with the "best case" difference of 0.75 bits occurring between $F_{1(CU)}$ and $F_{1(Shannon)}$. The pronounced concentration of the majority of interval tokens within those types ranked 1st ("-2"), 2nd ("0"), and 3rd ("+2"), further highlights the differences between letters and intervals (Figure 6-3). Given the totality of the consistent pattern of differences between the informetric properties of letters and intervals, the hypothesized equivalency between letters and intervals must be rejected.

The classification process was the largest contributor to the loss of information, with the amount of information lost being inversely proportional to the size of the classificatory sets. This finding was expected; however, the slight difference between the $F_{1(C15)}$ and $F_{1(CU)}$ was much less than one would expect given the significant differences in classificatory set sizes. We believe that this anomaly be can explained by the genre of music contained within the collection under investigation, and that it is not reflective of the classification process, *per se*.

The n-gramming process has a negligible effect on information loss.

If one takes the differences between the $F_{1(Shannon)}$ value of 4.14 bits and the $F_{1(n\text{-}grammed)}$ values for each of the n-grammed databases as a potential baseline predictor of possible retrieval performance under an FBIR model, certain trends emerge. Based solely on these limited data, it appears that *C3* databases are the weakest candidates for effective retrieval with an average difference, $F_{1(Shannon)} - F_{1(C3)}$, of 2.62 bits (4.14 - 1.52 = 2.62 bits, rounded). The *C7* databases similarly have an average difference, $F_{1(Shannon)} - F_{1(C7),}$ of 1.41 bits (4.14 - 2.73 = 1.41 bits, rounded), suggesting marginal retrieval performance. The *C15* databases appear to be stronger candidates with an average $F_{1(Shannon)} - F_{1(C15)}$ difference of 0.86 bits (4.14 - 3.28 = 0.86, rounded). The strongest candidates are the *CU* databases with an average $F_{1(Shannon)} - F_{1(CU)}$ difference of 0.75 bits. Furthermore, given the slight difference between average $F_{1(C15)}$ and $F_{1(CU)}$ values (0.86-0.75 = 0.11 bits) one would expect only a slight, perhaps insignificant, difference in performance between these two classes of databases. We will revisit the efficacy of these predictions throughout the remainder of this thesis.

6.3.3 Descriptive n-gram data and analyses

Taking the mean number of n-gram tokens per song as 50 tokens/song (Table 6-5), suggests that there might be some kind of equivalency between a folksong and a textual record's long title, short abstract, or keyword list. Just as we see in the range of interval tokens (Table 6-2), there is a wide range of n-gram tokens per song. In fact, since the number of n-gram tokens is simply a function of the choice of n-gram length, little more need be said about the descriptive data concerning n-gram tokens other than to re-emphasize the notion of representational compactness. In this case, that the contents

of 9354 folksongs can be represented using only 475,000 tokens (approximately) is worthy of note.

Table 6-5. Descriptive data about n-gram tokens

	L4	L5	L6
Number Tokens/Database	475,024	465,670	456,316
Mean Tokens/Song	50.78	49.78	48.78
SD Tokens/Song	29.37	29.37	29.37
Maximum Tokens/Song	498	497	496
Minimum Tokens/Song	4	3	2

Table 6-6. Number of n-gram types

	C3	C7	C15	CU
L4	81 (81)	2,298 (2,401)	13,273 (20,736)	21,796 (3,111,696)
L5	243 (243)	12,622 (16,807)	50,954 (248,832)	64,902 (1.31E+08)
L6	729 (729)	50,730 (117,649)	126,346 (2,985,984)	139,428 (5.49E+09)
	Actual count of types (Theoretic Maximum)			

As stated before, with any classification process the amount of information lost through the collapsing of data into categories is inversely proportional to the number of categories available. With this in mind, it is possible to envision the *C3*, *C7*, *C15*, and *CU* sets as lying upon an information-loss continuum, with *C3* representing "high" information loss, *C7* "medium" information loss, *C15* "low" information loss, and *CU* "no" information loss. Interacting with the classification process, is the n-gramming process which can also contribute to information loss by limiting the number of possible types because the number of potential unique strings (i.e., n-gram types) varies exponentially with the length of the n-gram according to the equation:

$$\text{Theoretic Maximum of Types} = C^{L}$$

where *C* is the size of the classificatory set, and, *L* the n-gram length.[48]

Table 6-6 clearly demonstrates how the interaction of the classification and n-gramming processes creates the information-loss continuum as represented by the remarkable range in the number of n-gram types present in each of the n-grammed databases. Note especially that the number of n-gram types found in each database does not increase in a linear manner as one moves from *C3L4* to *CUL6*. The pattern of value changes found in Table 6-6 actually reflects the $CLASS^{NLEN}$ relationship. Thus, for example, because $7^6 > 15^4$, (50,730 types (*C7L6*)) > (13,273 types (*C15L4*)). This pattern, dependent on the $CLASS^{NLEN}$ relationship, can be found throughout the remaining analyses of n-gram types.

Table 6-7. Descriptive data concerning n-gram types

	C3L4	C3L5	C3L6	C7L4	C7L5	C7L6	C15L4	C15L5	C15L6	CUL4	CUL5	CUL6
Mean Types/Song	26.19	32.75	36.62	38.04	40.12	40.97	39.06	40.68	41.33	39.20	40.77	41.40
SD Types/Song	8.65	13.15	16.83	16.90	19.67	21.30	17.87	20.22	21.67	18.03	20.32	21.75
Maximum Types/Song	70	129	190	192	250	285	209	260	300	212	260	303
Minimum Types/Song	4	3	2	4	3	2	4	3	2	4	3	2
Mean (per Song) Tokens/Type	1.94	1.52	1.33	1.33	1.24	1.19	1.30	1.22	1.18	1.30	1.22	1.18
Mean Songs/Type	2723.73	1046.51	357.39	52.03	10.42	3.53	6.07	2.89	1.79	3.51	2.27	1.66
Mean (per Type) IDF	1.78	3.16	4.71	7.49	9.81	11.37	10.59	11.66	12.35	11.38	12.01	12.46
Δ Mean IDF ($IDF_{(MAX)}$-$IDF_{(Mean)}$)	11.41	10.03	8.48	5.70	3.38	1.82	2.60	1.53	0.84	1.81	1.18	0.73

The number of n-gram types in each database (i.e., the size of the n-gram "vocabularies") has important implications. The number of types present in the *C3* databases is remarkably low (81 types (*C3L4*), 243 types (*C3L5*), and 729 types (*C3L6*)), especially given the 9354 songs in the collection. More significant is that the *C3* databases have "saturated", or used up, all the available n-gram types (i.e., the number of

[48] Using our study taxonomy, this can also be expressed as:

$$\text{Theoretic Maximum of Types} = CLASS^{NLEN}.$$

n-gram types present equals the theoretic maximum for that combination of CLASS and NLEN).[49] In an FBIR system that uses an inverted-file approach, having a "saturated" vocabulary causes the length of the postings lists to grow but not the length of the dictionary file. Such a growth pattern quickly undermines retrieval performance from both the efficiency (i.e., speed), and effectiveness (i.e., precision) standpoints. Within the context of the *C3* databases, for example, the presence of only 81 (*C3L4*), 243 (*C3L5*), or 729 (*C3L6*) "terms" (i.e., n-gram types), with which to describe all the songs, and then with which to retrieve a given song, all but assures that retrieval performance will be unsatisfactory. Furthermore, any growth in the number of songs indexed can only lead to poorer performance. Under the logic of this brief analysis, it appears that *C7L6* might be the minimally acceptable combination of CLASS and NLEN.

Examination of the descriptive data concerning n-gram types (Table 6-7) provides some further insights into the attributes and potential utility of the various database configurations. The range of values found in the **Mean Types/Song** row progresses, as one would expect, *generally*, from a minimum of 26.19 types/song (*C3L4*) to a maximum of 41.40 types/song (*CUL6*).[50] We qualify the preceding statement because of the aforementioned influence of the $CLASS^{NLEN}$ relationship.[51]

By comparing the two extreme values (i.e., 41.40-26.19=13.11), we can see the range of effect on information loss that the application of the n-gramming and classification processes have on our data. Thus, on average, 13.11 types per song are "destroyed" by application of the *C3L4* scheme instead of the *CUL6* scheme. This difference of 13.11 types/song represents, on average, the loss of 13.11 potential access points, or index entries, ***for each song***; a substantial loss of information given the relatively modest average length of the folksongs being indexed.

---

[49] *C7L4* and *C7L5* are also very close to saturation.

[50] Given the abundance of data found in Table 6-7 and Table 6-10, we will limit our demonstration of the key concepts to the extreme values. The concepts that hold in our examination of the extremes also hold for the intermediate values, only to a lesser extent.

Within the context of our hypothesized equivalency between n-grams and words, these values again suggest an equivalency between a folksong and a long, descriptive title, like those found in scientific articles and/or a short abstract. Seen as a list of "keywords" or "descriptors," the mean number of n-gram types/song represents a rather exhaustive level of applied indexing. For the *C3L4* database the list of "keywords" or "descriptors" for each song would be, on average, 26 terms long, and for the *CUL4* database, 41 terms long. When compared to a selection of the standard test collections, the music databases have a mean types/song that is greater than the mean types/record for the NPL, CACM, and INSPEC collections (*C3L4* being the one exception) (Table 6-8). Furthermore, Burgin (1991) reports strong retrieval performances from his evaluation of various representations of a cystic fibrosis collection that have types/record means that range from 49.5 types/record to 27.9 types/record. Below this range, he notes performance degradation. Only *C3L4* (26.19 types/song) falls below Burgin's threshold. These findings provide evidence that our music databases, when searched using text retrieval methods, should have retrieval characteristics comparable to the standard test collections (i.e., the number of searchable types/song is not pathologically low).

---

[51] The influence of the CLASS^{NLEN} relationship holds for the remainder of the data in Table 6-7 and Table 6-10 and will be mentioned again only intermittently.

Table 6-8. Music databases and standard test collections compared: Mean Types/Record[52]

		NPL	CACM	INSPEC	CISI	MED	CRAN
	Mean Types/Record	19.96	24.52	32.5	46.55	51.6	53.13
C3L4	26.19	*6.23*	*1.67*	*-6.31*	*-20.36*	*-25.41*	*-26.94*
C3L5	32.75	*12.79*	*8.23*	*0.25*	*-13.8*	*-18.85*	*-20.38*
C3L6	36.62	*16.66*	*12.1*	*4.12*	*-9.93*	*-14.98*	*-16.51*
C7L4	38.04	*18.08*	*13.52*	*5.54*	*-8.51*	*-13.56*	*-15.09*
C7L5	40.12	*20.16*	*15.6*	*7.62*	*-6.43*	*-11.48*	*-13.01*
C7L6	40.97	*21.01*	*16.45*	*8.47*	*-5.58*	*-10.63*	*-12.16*
C15L4	39.06	*19.10*	*14.54*	*6.56*	*-7.49*	*-12.54*	*-14.07*
C15L5	40.68	*20.72*	*16.16*	*8.18*	*-5.87*	*-10.92*	*-12.45*
C15L6	41.33	*21.37*	*16.81*	*8.83*	*-5.22*	*-10.27*	*-11.80*
CUL4	39.20	*19.24*	*14.68*	*6.70*	*-7.35*	*-12.40*	*-13.93*
CUL5	40.77	*20.81*	*16.25*	*8.27*	*-5.78*	*-10.83*	*-12.36*
CUL6	41.40	*21.44*	*16.88*	*8.90*	*-5.15*	*-10.20*	*-11.73*
ITALIC = Mean Types/Song – Mean Types/Document							

The range of values found in the **Mean (per Song) Tokens/Type** (Table 6-7) row regresses, *generally*, from a maximum of 1.94 tokens/type (*C3L4*) to a minimum of 1.18 tokens/type (*CUL6* and *C15L6*).[53] This is a measure of the average amount of redundant information found in each song. Thus, in each *C3L4* song, on average, there are almost 2 tokens representing each type present, while in *CUL6* there is closer to 1 token per type. Redundancy is not necessarily bad; however, one must be clear as to its cause. Within a given song, having several tokens that represent a particular type might be an indication of the importance of that particular type as an expression of thematic, or motivic, material. In IR theory, this is the "term frequency," or *tf*, measure of importance where the repetition of a type (i.e., word) within a record is taken as a possible indicator of the subject matter, or "aboutness," of that record. By examining the difference between the extreme values (1.94 - 1.18 = 0.76), we see that *at least* 39% of the redundancy found in the *C3L4* database (0.76/1.94 * 100 = 39%) is not attributable to thematic repetition; rather, the redundancy is attributable to the collapsing of otherwise distinct n-grams into broader classes. This implies that the application of a *tf* weighting scheme to those

[52] Data taken from Salton and Buckley (1988).

[53] Before rounding the values for presentation purposes, *CUL6* < *C15L6*.

databases where redundancy is artificially enhanced could have diminishing efficacy (i.e., will not necessarily increase precision) in proportion to the amount of redundancy created by the n-gramming and classification processes. However, the collapsing of types into broader classes could improve recall, so one must be cautious in one's deliberations. If high recall were a design imperative, the higher levels of within-song redundancy could be seen as a positive attribute.

Table 6-9. Music databases and standard test collections compared: Mean (per record) Tokens/Type[54]

		NPL	CACM	INSPEC	CISI	MED	CRAN
	Mean (per record) Tokens/Type	1.21	1.35	1.78	1.37	1.54	1.58
C3L4	1.94	*0.73*	*0.59*	*0.16*	*0.57*	*0.40*	*0.36*
C3L5	1.52	*0.31*	*0.17*	*-0.26*	*0.15*	*-0.02*	*-0.06*
C3L6	1.33	*0.12*	*-0.02*	*-0.45*	*-0.04*	*-0.21*	*-0.25*
C7L4	1.33	*0.12*	*-0.02*	*-0.45*	*-0.04*	*-0.21*	*-0.25*
C7L5	1.24	*0.03*	*-0.11*	*-0.54*	*-0.13*	*-0.30*	*-0.34*
C7L6	1.19	*-0.02*	*-0.16*	*-0.59*	*-0.18*	*-0.35*	*-0.39*
C15L4	1.30	*0.09*	*-0.05*	*-0.48*	*-0.07*	*-0.24*	*-0.28*
C15L5	1.22	*0.01*	*-0.13*	*-0.56*	*-0.15*	*-0.32*	*-0.36*
C15L6	1.18	*-0.03*	*-0.17*	*-0.60*	*-0.19*	*-0.36*	*-0.40*
CUL4	1.30	*0.09*	*-0.05*	*-0.48*	*-0.07*	*-0.24*	*-0.28*
CUL5	1.22	*0.01*	*-0.13*	*-0.56*	*-0.15*	*-0.32*	*-0.36*
CUL6	1.18	*-0.03*	*-0.17*	*-0.60*	*-0.19*	*-0.36*	*-0.40*
ITALIC = Mean (per song) Tokens/Type – Mean (per document) Tokens/Type							

When compared to some of the standard test collections, our mean (per song) tokens/type data again provide support for our hypothesized equivalency between n-grams and words. The strongest similarities can be found between our databases and the NPL, CACM, and CISI test collections (Table 6-9). These similarities suggest that the application of text retrieval methods (i.e., term-weighting schemes) that work well for these test collections should also work well for the music databases. Salton and Buckley (1988) report upon a comprehensive examination of an extraordinary variety of weighting schemes and their efficacy vis-à-vis the six test collections used in our comparisons. Our choice of a term-weighting scheme for use in the Phase II retrieval

[54] Data taken from Salton and Buckley (1988).

evaluations is based upon these similarities and the findings of Salton and Buckley (1988). Chapter 6.4 explicates this in greater detail.

Having examined the within-song (i.e., *tf*) characteristics of the n-grams, we now turn our attention to the across-song characteristics of the n-gram types. In IR theory, the number of documents in which a given term is found is known as the "document frequency," or *df*, of that term. In general, those terms with lower *df* values are considered to be the better candidates for use in indexing.[55] In the row labeled **Mean Songs/Type** (Table 6-7), the values presented are the means of all the *df* values for each database. Such a measure provides a broad-based indicator of the potential utility of the collection of types as index entries (i.e., the quality of each collection of types as potential "indexing vocabularies"). Again, we see the now-familiar regression of values from the maximum of 2723.73 songs/type (*C3L4*) to the minimum of 1.66 songs/type (*CUL6*). As absolute measures, these data are quite telling. They tell us that, on average, each n-gram type in the *C3L4* database participates in 2724 songs (rounded) while in *CUL6* each type can be found in only 2 songs (rounded). In an inverted file, the average length of the postings list for *C3L4* would be 2724 items, and for *CUL6*, 2 items. Thus, if one were to randomly select any n-gram type from the *C3L4* database, and then submit it as a query against that database, one would expect to retrieve, on average, 2724 songs (i.e., 29% of the database), a stunningly dismal performance by any measure. The same test performed on the *CUL6* database would return a remarkably small set of 2 songs, on average. Seen relative to one another, these values, and the potential utility that they represent, are an overwhelming three orders of magnitude apart ($2.72 * 10^3 / 1.66 * 10^0 = 1.60 * \mathbf{10^3}$). This disparity of three orders of magnitude also holds true for the values averaged within each of the *C3* and *CU* databases: the *C3* databases have an average 1378.88 songs/type and the *CU* databases, 2.48 songs/type. The *C7* databases have an average of 22.00 songs/type. While one order of magnitude greater than *CU*'s average of 2.48 songs/type, the absolute value of 22.00 songs/type for *C7* represents a marginally

---

[55] More precisely, it is the *df* value *relative to* the size of the collection of documents that is held to determine the potential utility of a term. More about this notion of relativity follows in our discussion of the Inverse Document Frequency (IDF) measure.

acceptable set size should one perform the retrieval test mentioned above.[56] We also note that there is only a slight difference between *CU*'s average 2.48 songs/type, and *C15*'s 3.58 songs/type. Thus, within the constraints of this singular analysis, we again see that *C3* appears unsuitable, and *C7* marginally suitable, for use in a MIR system. We also see that there might be only marginal differences in retrieval performance between the *C15* and *CU* databases.

The Inverse Document Frequency (IDF) of a term *j* is defined as:

$$IDF_j = \log_2 \frac{N}{n_j}$$

where N is the number of documents in the collection, and n_j the number of documents in which term *j* appears. Sparck-Jones (1972) developed the IDF metric as a method to automatically determine the "specificity" of a term, that is, the ability of a term to identify a specific document, or small set of documents. The maximum IDF value for a term in a collection of N documents is calculated as $\log_2 N/1$ (i.e., a term appears in only 1 of N documents. For our collection of 9354 documents (i.e., songs) the maximum IDF value is $\log_2 9354/1 = 13.19$. The minimum IDF value for a term in a collection of N documents is obtained when $N = n$ (i.e., a term appears in every document) so that $\log_2 N/n = 0$. As one can see, the IDF value is a relative measure of a term's potential utility as specifier of, or pointer to, a given document, or sub-set of documents. Those terms that appear in *all* documents have an IDF value of 0, thus clearly indicating their inability to specify, or point to, any document in particular, while those terms that point to a single document have the maximum IDF value for that collection.

We have taken the average IDF value for the n-gram types found in each of our databases as another broad-based indicator of the potential utility of the types as the basis for an "indexing vocabulary." In this case, the quality of the indexing vocabulary would manifest itself as the ability of the vocabulary to retrieve specific songs, and only those songs (i.e., high retrieval precision). Simply put, the collections of types contained

---

[56] In the IR literature, there is the often-cited value of the "30-document retrieved set" that is said to be the limit of acceptable set sizes for browsing returned records.

within the databases with **Mean (per Type) IDF** values closer to 0 are the inferior candidates for effective indexing (e.g., *C3L4* (1.78), *C3L5* (3.16), and *C3L6* (4.71)), while those closer to the collection maximum of 13.19 are the superior candidates (e.g., *CUL6* (12.46), *C15L6* (12.35), and *CUL5* (12.01)).

The information found in the **Δ Mean IDF** row is presented to show how close, or how far away, the mean IDF value for each database is from our particular maximum IDF value of 13.19. The pattern of values found in the **Δ Mean IDF** row is simply the converse of those found in the **Mean (per Type) IDF** row. Noteworthy, however, is proximity of the *CUL6* (0.73) and *C15L6* (0.84) values to the maximum IDF value, indicating the presence of a large number of types that occur in relatively few documents. This proximity also suggests very strong retrieval performance for these databases as measured by precision. Also, looking at the **Δ Mean IDF** values, averaged within each CLASS, the *C3* value (9.97) stands out again as distinctly inferior to the *C7* (3.63), *C15* (1.66) and *CU* (1.28) average values.

Table 6-10. Entropy data concerning n-grams

	C3L4	C3L5	C3L6	C7L4	C7L5	C7L6	C15L4	C15L5	C15L6	CUL4	CUL5	CUL6
Entropy (View A) Types/Collection	6.02	7.50	8.97	9.98	12.25	14.38	11.53	13.88	15.80	11.78	14.08	15.92
Entropy (View B) Types/Songs	6.18	7.63	9.07	10.06	12.31	14.43	11.66	13.97	15.88	11.94	14.19	16.01
Entropy ($H_{(Max)}$) Theoretic Max	6.34	7.92	9.51	11.17	13.62	15.63	13.70	15.64	16.95	14.41	15.99	17.09
Δ Entropy (ViewB-ViewA)	0.16	0.13	0.10	0.08	0.06	0.05	0.13	0.09	0.08	0.16	0.11	0.09
Δ Entropy ($H_{(Max)}$-ViewA)	0.32	0.42	0.54	1.19	1.37	1.25	2.17	1.76	1.15	2.63	1.91	1.17
Δ Entropy ($H_{(Max)}$-ViewB)	0.16	0.29	0.44	1.11	1.31	1.20	2.04	1.67	1.07	2.47	1.80	1.08
Δ Entropy (ViewA-Yavuz)	-3.70	-2.22	-0.75	0.26	2.53	4.66	1.81	4.16	6.08	2.06	4.36	6.20
Δ Entropy (ViewB-Yavuz)	-3.54	-2.09	-0.65	0.34	2.59	4.71	1.94	4.25	6.16	2.22	4.47	6.29

The range of values found in the **Entropy (View A)** row progresses from a minimum of 6.02 bits (*C3L4*) to a maximum of 15.92 bits (*CUL6*). Recall that View A is the *corpus* view, where each database is seen as a collection of types and their tokens (i.e., no partitioning by song). By examining the difference between the extreme values,

we see that 9.90 bits of information are lost, or "destroyed," (15.92 – 6.02 = 9.90 bits) by the application of the *C3L4* scheme instead of the *CUL6* scheme.

One way to highlight the magnitude of the 9.90 bit difference is to perform a small thought experiment. If one built a message system where there were 8 different, yet equally probable messages, that system would have an entropy of 3 bits because $\bar{H} = -\Sigma\ (1/8) \log_2 (1/8) = 3$ bits. Now imagine that each message is stuffed within its own envelope and that all the envelopes are placed within a lottery drum. Using our knowledge of entropy we can calculate the probability that one would retrieve from the drum an envelope which contained a particular message. Working backwards we see that a system entropy of 3 bits implies a 1 in 8 probability of randomly selecting a message because $2^{\mathbf{3}} = 8$. The key here is to realize that an entropy value of *x* bits implies a 1 in 2^x probability of a given message. Thus, in a 3 bit system, one would have a 1 in 8 chance of randomly selecting an envelope from the drum that contained a particular message. In a similar manner, imagine the collection of types found in each of our databases as a collection of "messages," where each type is a potential message. So, when we calculate the *C3L4* entropy to be 6.02 bits, we are stating that one would have 1 in 64.89 probability of randomly selecting from our database "drum" a particular "message," or n-gram type ($2^{6.02} = 64.89$). For the *CUL6* database, where entropy is 15.92 bits, the probability of selecting an "envelope" containing a particular "message," or n-gram type, would be 1 in 62,000.83 because $2^{15.92} = 62{,}000.83$. Thus, the 9.90 bit difference between *C3L4* and *CUL6* databases indicates that one is 955.43 times ($2^{9.90} = 955.43$) less likely to randomly select a particular *CUL6* type than a particular *C3L4* type. According to Shannon (Shannon and Weaver 1949), those systems with the least-predictable set of messages are those which are most informative. We therefore see that the 9.90 bit difference signifies a very large difference in information content between the *C3L4* and *CUL6* databases.

The **Entropy (View B)** row contains the entropy information for each database where only 1 token per type is used to denote the presence of a type within a song (i.e., the type counts are partitioned by song, with the within-song redundancy removed). The minimum View B entropy is 6.18 bits (*C3L4*) and the maximum 16.01 bits (*CUL6*). The

Δ Entropy (View B-View A) row illustrates the amount of information gained by removing the effect of within-song redundancy. That the values are rather low, (e.g., 0.16 bits (*C3L4*) and 0.09 bits (*CUL6*)) indicates the relatively small influence that within-song redundancy has on determining the average information value of an n-gram. However, that the values are positive indicates that the View B distributions of songs over types contain less variance than the View A distributions of tokens over types. This reduction in variance implies that the repetition of a given type within different songs is not constant. That is, for those types that occur more than once, some might occur twice in "Song *a*", once in "Song *b*", and thrice in "Song *c*", etc. This inconsistency in within-song repetition (i.e., the variance in *tf*) can be useful in differentiating "Song *a*" from "Song *b*" from "Song *c*". Thus, the positive values found in the **Δ Entropy (ViewB-ViewA)** row indicate that a retrieval scheme that employed a *tf* weighting component would have some efficacy.

The **Entropy ($H_{(Max)}$) Theoretic Max** row provides the theoretic maximum entropy for each database given the number of types found within the database. Recall that maximum entropy for a system is obtained when all type probabilities are equal (i.e., $H_{(Max)} = -\log_2(1/N)$, where N is the number of types). Note that maximum entropy for a system is also directly dependent on N (i.e., the number of types within that system). For example, a system with 2 types present has a maximum entropy of 1 bit because $-\log_2(1/2) = 1$, while a system with 64,000 types has a maximum entropy of 15.97 bits because $-\log_2(1/64{,}000) = 15.97$. The values calculated were obtained using the data found in Table 6-6 and again reflect the influence of the $CLASS^{NLEN}$ relationship. Thus, *C3L4*, with its 81 types, has the lowest maximum entropy (6.34 bits) while *CUL6* (139,428 types) has the highest maximum entropy (17.09 bits). On their own, these values are only important insofar as they represent the upper-limits of the amount of information that each database configuration can provide. For example, the maximum entropy for *C3L4* will ***never*** exceed 6.34 bits. Understanding that these values represent upper-limits allows for more meaningful comparisons between the various **Δ Entropy** measures presented in the bottom-half of Table 6-10.

The **Δ Entropy ($H_{(Max)}$-View A)** row contains the differences of the observed database entropies (View A) from their respective theoretic maxima ($H_{(Max)}$). The *C3* databases have the lowest differences with *C3L4* (0.32 bits), *C3L5* (0.42 bits), and *C3L6* (0.54 bits) for an average difference of 0.42 bits. The *CU* databases have the highest differences with *CUL4* (2.63 bits), *CUL5* (1.91 bits), and *CUL6* (1.17 bits) for an average difference of 1.90 bits. Recall from our discussion of interval entropy that the Zipfian conjunction of:

a) a few types that occur many times (i.e., the clustering of tokens about the highest-ranked types);

b) a steep downward slope as one progresses through the ranks; and,

c) a long tail of types that occur infrequently,

is the cause of the drop in entropy from the theoretic maximum to the observed value. In the *C3* case, the low average difference indicates the relatively equal distribution of tokens over the types (i.e., a distinctly non-Zipfian distribution). This relatively equal distribution is caused by the small number of types found within each of the *C3* databases. Because there are so few types relative to the number of tokens there are no types that occur very infrequently (i.e., once, or perhaps, twice). The frequencies of the most infrequently occurring types for the *C3* databases are 1925 (*C3L4*), 467 (*C3L5*), and 95 (*C3L6*). So as one progresses through the ranks, the slope is relatively slight, and constant. *C7L4* is the first database where there are types with a frequency of 1 (i.e., there are 65 types in *C7L4* that appear only once). It is also the first database that has a distribution of tokens over types that could be described as Zipfian. The 1.27 bit average difference between theoretic maxima and observed entropies for the *C7* databases is caused by the Zipfian shape of their distributions. This is also the case for the 1.69 bit and 1.90 bit average differences found in the *C15* and *CU* databases, respectively. That the difference values increase, from *C7* through *CU*, reflects the presence of increasingly-steeper slopes as one progresses from the highest-ranked n-gram through the upper 10 ranks, where the curves then flatten out on the way to their increasingly-long tails. For example, the slope for the top 10 ranked types in the *C7L4* database is 213.11 tokens/rank, while the same slope for *CUL4* is 303.89 tokens/rank. Furthermore, there

are 65 types that occur once in *C7L4*, while there are 76,067 singleton types in *CUL6*. Respectively, these values represent 2.82% (65 / 2,298 * 100 = 2.82) and 55% (76,067 / 139,428 * 100 = 54.56) of the number of types found in these databases. Thus, it is the complex interaction of steeper slopes and longer tails that causes the drop in entropy from theoretic maximum to the observed values.

The **Δ Entropy ($H_{(Max)}$-View B)** values are presented for information purposes only. The values are consistently lower in the **Δ Entropy ($H_{(Max)}$-View B)** than the **Δ Entropy ($H_{(Max)}$-View A)** row simply because the **Entropy (View B)** values are greater than the **Entropy (View A)** values. The key concepts brought forth in the analysis above also apply for these data.

The **Δ Entropy (View A-Yavuz)** and **Δ Entropy (View B-Yavuz)** rows represent the most fascinating of the entropy data. Recall that Yavuz estimated the entropy of English text as 9.72 bits (Yavuz 1974). When comparing the Yavuz value with our observed entropy values the partitioning of the observed values into two distinct groups is quite striking. The first group, composed solely of the *C3* databases, comprises those databases that have lower entropy values than text (i.e., negative differences). The second group, composed of the *C7*, *C15* and *CU* databases, comprises those databases that have entropy values greater than text (i.e., positive differences). The first inference one draws from these groupings is that one would expect a MIR system that used any of the *C3* databases to have poorer performance than a textual FBIR system of the same size (i.e., the same number of records). Similarly, it appears that the *C7*, *C15*, and *CU* databases should have retrieval performances, when compared to textual IR, ranging from relatively equal (i.e., *C7L4* with a difference of 0.26 bits) to markedly superior (i.e., *CUL6* with a difference of 6.20 bits). We also note that our *C3* databases will always have lower entropy values than text —even with the slight increase in entropy caused by the View B elimination of within-song redundancy— because the $H_{(Max)}$ values of 6.34 bits (*C3L4*), 7.92 bits (*C3L5*), and 9.51 bits (*C3L6*) are all less than text's 9.72 bits. Again, under the logic of this analysis, we see that the *C3* databases appear unsuitable, the *C7* databases appear acceptable, and the *C15* and *CU* databases appear promising, for use in a MIR system based upon the FBIR model.

Table 6-11. Number of songs in which the most frequently occurring n-gram type is found

	C3	C7	C15	CU
L4	7349 (79)	2294 (25)	1891 (20)	1891 (20)
L5	4742 (51)	982 (10)	719 (08)	719 (08)
L6	2603 (28)	403 (04)	311 (03)	311 (03)
	Number of Songs (Percentage of Database)			

Table 6-12. Top ranking types: text and n-grams compared

Rank	1st	2nd	3rd	4th	Top 10 (Cumulative)
Text	"the" 7.00%	"of" 3.64%	"and" 2.89%	"to" 2.61%	24.60%
C3L4	4.55%	3.97%	3.73%	3.41%	33.76%
C3L5	2.04%	2.03%	1.74%	1.70%	17.14%
C3L6	0.95%	0.88%	0.87%	0.86%	8.37%
C7L4	0.87%	0.77%	0.69%	0.67%	6.29%
C7L5	0.38%	0.29%	0.28%	0.22%	2.40%
C7L6	0.19%	0.12%	0.11%	0.10%	1.02%
C15L4	0.87%	0.63%	0.50%	0.41%	4.39%
C15L5	0.38%	0.21%	0.20%	0.16%	1.67%
C15L6	0.19%	0.09%	0.08%	0.07%	0.75%
CUL4	0.87%	0.63%	0.50%	0.41%	4.39%
CUL5	0.38%	0.21%	0.20%	0.16%	1.67%
CUL6	0.19%	0.09%	0.08%	0.07%	0.75%

Why is it that the *C7*, *C15*, and *CU* databases have higher entropy values than text? The answer lies in the shape of their distributions, in particular, the apparent absence of types that could be classed as "stopwords". Recall from our earlier discussion concerning stopwords (e.g., "the," "of," "to," and, "and," etc.) that stopwords are those terms that occur very frequently (i.e., both throughout a collection (View A) and within almost every document (View B)). In text, as the most frequently occurring types, the stopword terms form the pronounced head to the rank-probability distribution of word tokens over word types. In Table 6-11, we see that the relative document frequencies of the most frequently occurring n-gram types are remarkably low. For example, in *C7L4* the most frequently occurring type occurs in 2,294, or 25%, of the songs. For *CUL6*, the most frequently occurring type occurs in a scant 311, or 3%, of the songs. A type that occurs in only 3% of the songs can hardly be classed as a stopword. This absence of n-gram "stopwords" results in a relatively flat head to the n-gram distributions. Thus, when compared to text, the MusiFind databases have distributions with flatter heads (i.e.,

relatively fewer tokens that cluster about the highest-ranked types). See Figure 6-4 for the View A depiction of this phenomenon. With the exception of the *C3* databases, they also have relatively longer tails (i.e., a greater relative proportion of infrequently occurring types). Taken together, these two factors result in smoother, more equalized distributions. Because entropy increases as the distributions equalize, the *C7*, *C15*, and *CU* databases have greater entropy values than text.

Salton (1989) informs us that, in English, the four most frequently occurring word types, as determined from a sample of 1,000,000 word tokens, are "the", "of", "and", and "to" (Table 6-12). The most frequently occurring type, "the", accounts for approximately 7.0% of English tokens. The remaining three types, "of" (3.6%), "and" (2.9%), and "to" (2.6%), along with "the" (7.0%), account for approximately 16.2% of the tokens found in English text. In the *C3L4* database, the four most frequently occurring n-gram types account for 15.6% of the tokens present, a value not significantly different from text's 16.2%. There are two differences between English text and the C3L4 database worthy of note, as the data in Table 6-12, and the chart in Figure 6-4, clearly show:

1) the slope to the head of the *C3L4* distribution is markedly less pronounced than that of text; while,

2) the cumulative percentage of tokens accounted for by *C3L4*'s top 10 ranking types (33.8%), is greater than text's 24.6 %.

Thus, the *C3L4* distribution data give rise to an interesting situation with regards to the stopword issue. As we have shown in our earlier examination of the Table 6-6 data, the *C3* databases are anomalous in that they are "saturated" (i.e., they use all of their available types). In the *C3L4* case there are only 81 types available over which are distributed 475,024 tokens so the probability is quite high that each type would account for a substantial proportion of the tokens present.[57] Given that the relative proportion of the "head" types in *C3L4* (15.6%) and text (16.2%) are so similar, and given that the remaining 74 types in the *C3L4* database each account for a substantial proportion of its

---

[57] This is confirmed by the low 0.32 bit difference between *C3L4*'s observed and theoretic maximum entropy values.

tokens, one could make the argument that ***most*** of the n-gram types in the *C3L4* database should be treated as stopwords. For the remaining eleven databases, the flat shape and relatively small size of their distribution heads suggest that there are few, if any, n-gram types that are obvious candidates for inclusion on a stoplist, particularly in the *C7*, *C15*, and *CU* databases. For example, starting with the *C7L4* database, the maximum proportion of occurrence of the most frequently occurring type is 0.87% (*C7L4*) and the minimum 0.19% (*CUL6*). The data found in the **Top 10 (Cumulative)** column reinforces our contention that there is no obvious stopword issue for the *C7*, *C15* and *CU* databases, with *C7L4* having the maximum cumulative proportion of 6.29% and *CUL6* (and *C15L6*) having the astoundingly-low minimum cumulative proportion of 0.75%.

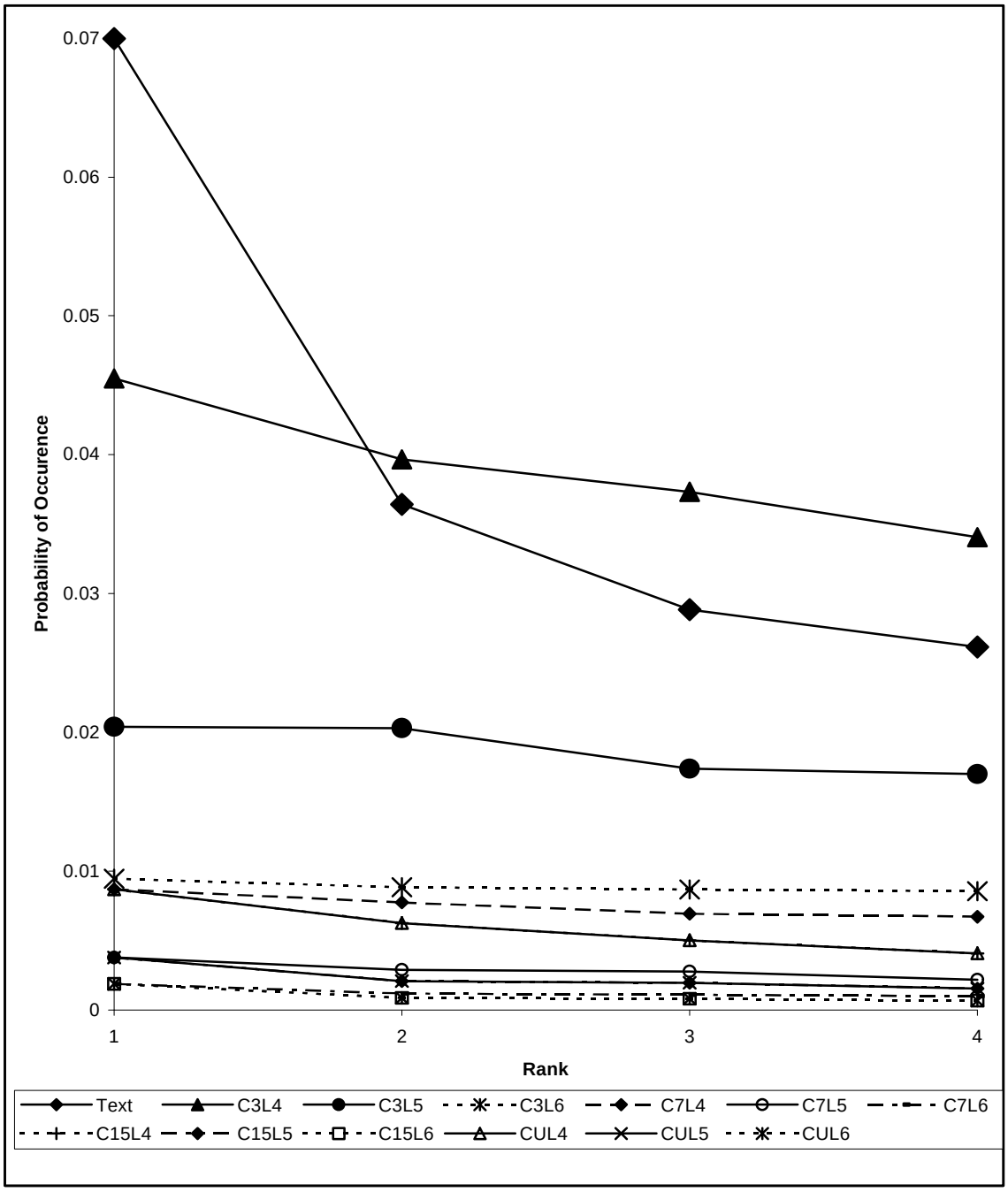


Figure 6-4. Distribution heads compared: text and music n-grams (View A)

The data presented in Table 6-11 can be used to model the "worst-case" retrieval performance of the databases. Here we define "worst-case" retrieval performance as the number of songs returned by a database's most frequently occurring n-gram type if that n-gram type were submitted as a query. That is, *C3L4* is the "worst-of-the-worst" with 7349, or 79%, of the songs returned. *C15L6* and *CUL6* are the "best-of-the-worst" with

311, or 3%, of the songs returned. *C7L5* and *C7L6*, with respectively 982 (10%) and 403 (4%) of the songs returned, appear to have better than acceptable projected retrieval performances.

Why does the "worst-case" performance matter? These data were calculated as a back-up measure should the Phase II retrieval experiments reveal that the MusiFind approach to MIR is unworkable. Instead of abandoning the n-gramming of interval-only melodic information altogether as a means of representing, and then retrieving songs, we envisioned the possible utility of a more sophisticated two-staged approach. The first stage would use the MusiFind approach to identify the set of songs that *loosely* matched a given query. The second stage would employ one of the sophisticated approximate string-matching algorithms (e.g., McNab et al. 1997) to perform a more precise search on the set of songs passed to it by the first-stage MusiFind search. Recall that these approximate string-matching methods are essentially linear scans of the melodic strings and, as such, their performance degrades as the number of melodic strings processed increases. By examining the worst-case performance of the MusiFind approach, we have determined the maximum size of a set returned by our hypothetical first stage that then would be passed on to the second, more computationally expensive, stage. Thus, we see from these worst-case data that the MusiFind approach, if it were to fail as a primary means of retrieval, at least shows strong promise as an efficient first-stage filter.

The data presented in Table 6-13 can be used to model the "best-case" retrieval performance of the databases. Here we define "best-case" retrieval performance as the probability that a randomly selected n-gram would occur in 20 or fewer songs, given that it is to be found among the database's types. The number 20 was selected as representing a reasonable set size for browsing. *C3* is not to be found in Table 6-13 as the "best" n-gram type (from *C3L6*) would return 83 songs if submitted as a query. Not surprisingly, *CUL6* has the superlative projected performance with a 0.62 probability of returning only 1 song. With a remarkable 0.98 probability of a randomly selected n-gram returning 20 or fewer songs, *CUL6* should perform quite well indeed. The *C15* databases have probabilities slightly lower than those of the *CU* databases, suggesting that the *C15* retrieval performances should be just below that of the *CU* databases. The *C7* data again

place its databases in a grey area: the *C7L4* projections are less-than-impressive, while the *C7L6* data are, in fact, superior to the *C15L4* data.

Table 6-13. Probability that a given n-gram occurs in *x* or fewer songs

X	C7L4	C7L5	C7L6	C15L4	C15L5	C15L6	CUL4	CUL5	CUL6
1	0.03	0.13	0.29	0.26	0.40	0.57	0.40	0.50	0.62
2	0.06	0.21	0.44	0.37	0.56	0.73	0.53	0.65	0.77
3	0.09	0.27	0.55	0.45	0.65	0.81	0.61	0.73	0.84
4	0.11	0.32	0.62	0.50	0.70	0.86	0.66	0.78	0.88
5	0.13	0.36	0.68	0.54	0.75	0.89	0.70	0.81	0.91
6	0.14	0.39	0.72	0.57	0.78	0.91	0.72	0.84	0.92
7	0.16	0.42	0.75	0.60	0.80	0.92	0.75	0.85	0.94
8	0.17	0.45	0.78	0.62	0.82	0.94	0.76	0.87	0.95
9	0.19	0.47	0.80	0.64	0.84	0.94	0.78	0.88	0.95
10	0.20	0.50	0.82	0.66	0.85	0.95	0.79	0.89	0.96
15	0.25	0.58	0.88	0.72	0.90	0.97	0.84	0.93	0.98
20	0.28	0.65	0.92	0.76	0.92	0.98	0.86	0.95	0.98

We must note here that the Table 6-13 data represent the probabilities of retrieving *x* songs given the submission of a *single* n-gram. In the case of *L4* databases, this would be the equivalent of submitting a single four-letter word as a query to an FBIR system. The probabilities of retrieving *x* songs for a given query could be equal, but most likely greatly superior, if one were to submit as queries two, or more, randomly selected n-grams, conjoined with a Boolean "AND." In actual practice, this scenario is more likely than not. For example, in the case of the *L4* databases, a short query of 6 intervals would be n-grammed into three n-grams of length-4 (i.e., three 4-grams). If these n-grams were "ANDed" before submission, the probability that the set of retrieved songs would be a manageable size would be greatly enhanced, particularly in the case of the *C7L4* database.

6.3.4 Term discrimination data and analyses

The term discrimination data presented in Table 6-14, Figure 6-5 and Figure 6-6 provide yet another set of broad-based indicators of potential retrieval performance. With regard to the data in Table 6-14 and Figure 6-5, a certain amount of analytic caution must be applied. There are two sources of error of which one should be cognizant. First,

in order to complete the computation of the individual DV_k values for each database within an acceptable amount of time, the *exact centroid method* (Crouch 1988) used in Dubin's *TDV.pl* generates approximate, not precise, discrimination values.[58] Second, given the large number of types in the larger databases (e.g., *C15L6* and *CUL6*), and given that PERL does not handle floating-point operations with extended precision, a small amount of rounding error (i.e., at the 12^{th} decimal place) has been detected. With these caveats in mind, it is best not to interpret the discrimination data as absolute values. However, as relative measures, that is, comparing relative proportions of positive and negative discriminators between the databases, the discrimination data are perfectly valid.

Table 6-14. Term discrimination data

	C3	**C7**	**C15**	**CU**
L4	70 (11)	2,169 (129)	13,083 (190)	21,601 (195)
L5	218 (25)	12,183 (439)	50,185 (769)	64,054 (848)
L6	673 (56)	49,204 (1526)	122,959 (3,387)	135,645 (3,783)
	Number of positively discriminating n-gram types (Number of negatively discriminating n-gram types)			

[58] The precise calculation of DV_k for each term in a database has $O(2Knt)$ complexity, where K is the average number of types per document, n the number of documents and, t the number of types in the database (Salton 1975). Notwithstanding the substantial improvement in computational complexity afforded by the exact centroid method, the DV_k values for *CUL6* took approximately 18 hours to calculate using a 290 Mhz. Intel PII PC running Linux with 128 M of RAM.

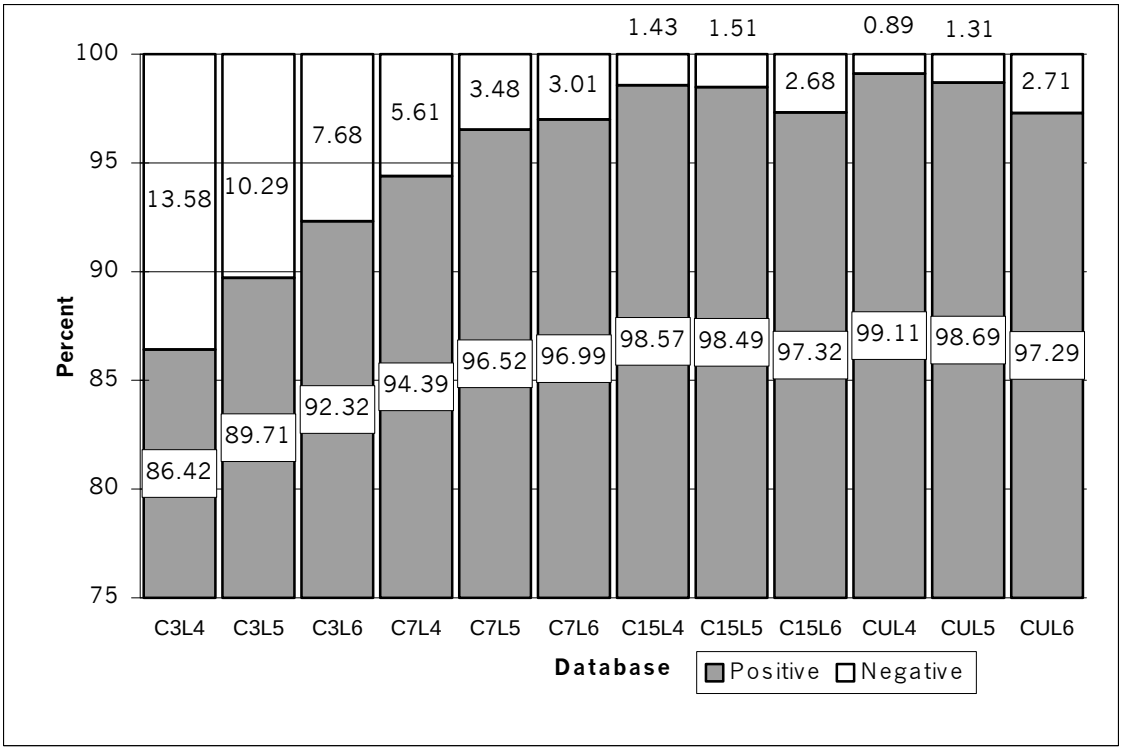


Figure 6-5. Percentage of positive and negative discriminators

The proportion of positive discriminators ranges from a minimum of 86.42% (*C3L4*) to a maximum of 99.11% (*CUL4*). Within the *C15* and *CU* databases, there is a slight drop in the proportion of positive discriminators as NLEN increases. In a *post hoc* analysis of the discrimination data associated with the *C15L5*, *C15L6*, *CUL5*, and *CUL6* databases, we discovered a number of low-frequency terms (i.e., document frequencies of 1 and 2) that have negative discrimination values. However, the DV_k values for these low-frequency terms are so close to 0 (i.e., negative at the 12^{th} decimal place) that we suspect rounding error to be the principal cause of the decrease in the proportion of positive discriminators. Furthermore, Salton (1975) informs us that terms with document frequencies approaching 1 generally have positive DV_k values approaching, or equal to, 0.

It is the examination of the averages within each database CLASS that provides us with the most meaningful information. The average proportion of negative discriminators for each CLASS are 10.52% (*C3)*, 4.03% (*C7*), 1.87% (*C5*), and 1.64% (*CU*). Again, we see that the databases can be partitioned into three groups: *C3* (Group 1), *C7* (Group 2), *C15* and *CU* (Group 3). Since retrieval performance, as measured by

precision, is inversely proportional to the relative number of negative discriminators present in a database, we can also rank these databases according to their expected retrieval performances from worst (*C3*), to better (*C7*), to best (*C15* and *CU*).

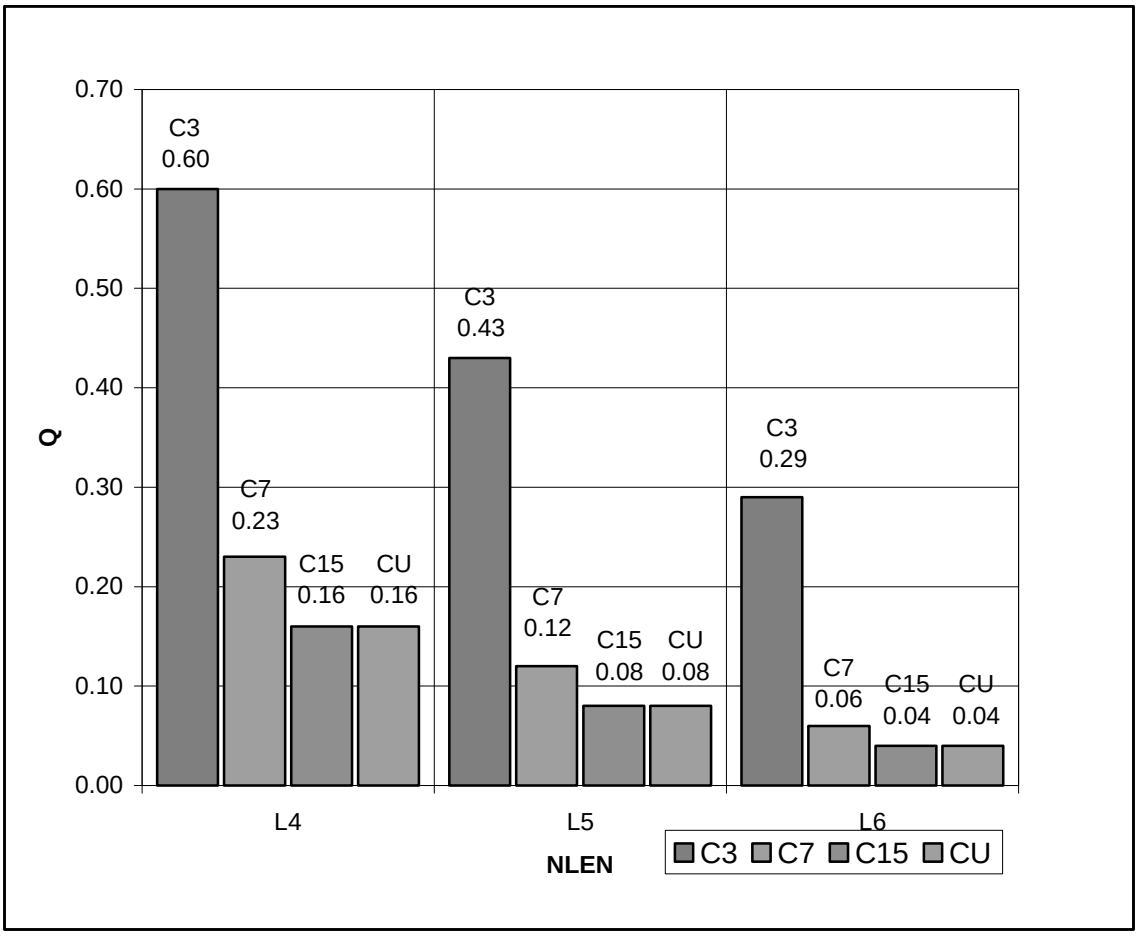


Figure 6-6. Document space density *Q* values for each database

Nowhere else is the partitioning mentioned above more clearly delineated than in the *Q* value data presented in Figure 6-6. The average *Q* values for each CLASS are 0.44 (*C3*), 0.14 (*C7*), and 0.09 (*C15* and *CU*). The extreme *Q* values range from a maximum of 0.60 (*C3L4*) to a minimum of 0.04 (*C15L6* and *CUL6*). Recall that *Q* is the average similarity between documents as measured by the cosine correlation metric. Also recall that the cosine correlation metric has a value of 0 when there is no similarity between documents and a value of 1 when the documents are identical. Thus, the *C3* databases, with a mean *Q* value of 0.44, have very poor between-document discrimination, making the precise retrieval of desired songs very unlikely. Similarly, the *C15* and *CU* databases,

with their identical mean *Q* value of 0.09, should provide the opportunity for highly precise retrieval. The *C7* data (0.14) suggest relatively good retrieval precision, especially *C7L5* (0.12) and *C7L6* (0.06). It is difficult, however, to ascertain from these data whether *C7L4*, with its *Q* value of 0.23, is discriminating enough for effective retrieval precision.

6.3.5 Descriptive n-gram data and analyses: concluding remarks

Before moving on to the informetric modeling section of this chapter, we should summarize the general themes that we see emerging from our examination of the descriptive data concerning n-grams. The five principal themes are:

1) The variance of the informetric data is primarily influenced by the $CLASS^{NLEN}$ relationship. Since CLASS is the base of this relationship, and there is a much wider range of values associated with CLASS (i.e., 3, 7, 15, and 42 (*CU*)), CLASS plays the greater role in shaping the informetic properties of the n-grams.

2) By every measure (e.g., type counts, songs per type, IDF, entropy, potential stopwords, negative discriminators, *Q* values, etc.), the *C3* CLASS of databases have distinctly inferior informetric characteristics. The *C3* data indicate that use of any of these databases in a MIR system would be unprofitable.

3) The *C7* databases have informetric properties that place them in a kind of analytic "grey zone". For example, from the entropy analyses, the *C7* databases appear quite acceptable (i.e., all have entropies greater than that of text); but, from the term saturation standpoint, *C7L4* and *C7L5* appear problematic. For another example, *C7L4* has rather weak "best case" probabilities, while *C7L6*'s are promising. In *toto*, the *C7* data suggest that the *C7* databases represent the lowest, if at all, acceptable $CLASS^{NLEN}$ combination.

4) The *CU* databases are consistently superior from an informetric viewpoint. However, the margin of superiority evident in the *CU* data over the *C15* data is consistently slight. Thus, the *CU* and *C15* data suggest promising retrieval

performance from members of these database CLASSes but little, if any, performance difference between the two.

5) The *C7*, *C15* and *CU* databases have informetric characteristics that suggest that these music databases would have retrieval performances, when compared to text databases, that range from similar (e.g., *C7L4*) to greatly superior (e.g., *CUL6*).

With regard to our hypothesized equivalency between n-grams and words, the descriptive n-gram data lead us to a qualified acceptance of the notion. As a point of analytic departure, the hypothesized equivalency has proven itself too fruitful a metaphor to be abandoned. For example, it has led us to uncover some important similarities between our music databases and some of the standard test collections (e.g., CACM, NLP, and CISI). It has also led us to uncover the consistently poor informetric properties of the *C3* databases. It has shown us that there is no obvious informetric reason why the application of text retrieval methods should not work for the *C7*, *C15*, and *CU* databases. The acceptance of equivalency is less justified, however, when one tries to map one-to-one some of the individual informetric properties of n-grams and words. For example, (ignoring the *C3* case) the entropy values of the n-grams are consistently greater than that of text. For another example, there does not appear to be a group of n-grams that one would classify as stopwords. Also, the within-song tokens/type for the music databases is generally lower than the within-document tokens/type of text. So, in conclusion, we continue to assert an equivalency between music n-grams and words, but we acknowledge that equivalency is primarily pragmatic and metaphoric, rather than explicitly delineated.

Because of the overwhelming informetric evidence that the *C3* databases would have distinctly inferior retrieval performances, we concluded that further examination of the *C3* databases would be unwarranted. Therefore, for the remainder of the study (i.e., the informetric modeling, and the Phase II IR experiments) only the *C7*, *C15*, and *CU* databases were evaluated.

6.3.6 Informetric modeling

Of the four distribution models studied –Zipf, Mandelbrot-Zipf (MZ), Zero-truncated Generalized Waring (GW), and Zero-truncated Generalized Inverse Gaussian-Poisson (GIGP)– only one, the GIGP, successfully fits any of the observed distribution data (at $p = 0.05$, and the appropriate degrees of freedom for the data). The GIGP distribution fits five of the View A representations (Table 6-15) and three of the View B (Table 6-16).[59] For all the representations save three, the GIGP provides the "best fit" (i.e., lowest C^2 values, whether statistically significant or not) (Table 6-17). The GW distribution is the best fit for *C7L6* (Views A and B) and *C15L5* (View A) albeit none at a statistically significant level. The Zipf and MZ distributions are consistently the poorest-fitting distribution models.

Table 6-15. GIGP (zero-truncated) model fitting data: View A

	α	θ	γ	X^2	d.f.	Critical X^2	Decision (p=0.05)
C7L4	0.14	0.9976	0.42	262.87	275	315	**ACCEPT**
C7L5	0.25	0.9897	0.24	387.94	240	277	REJECT
C7L6	0.57	0.9707	-0.09	829.76	125	152	REJECT
C15L4	0.42	0.9970	-0.17	318.16	283	323	**ACCEPT**
C15L5	0.48	0.9889	-0.36	500.89	182	214	REJECT
C15L6	0.89	0.9804	-0.78	123.59	108	133	**ACCEPT**
CUL4	0.41	0.9981	-0.37	294.17	279	319	**ACCEPT**
CUL5	0.39	0.9914	-0.50	370.59	210	245	REJECT
CUL6	0.74	0.9832	-0.82	114.44	110	135	**ACCEPT**

Table 6-16. GIGP (zero-truncated) model fitting data: View B

Scheme	α	θ	γ	X^2	d.f.	Critical X^2	Decision (p=0.05)
C7L4	0.23	0.9970	0.41	307.89	254	292	REJECT
C7L5	0.73	0.9901	0.14	300.40	227	263	REJECT
C7L6	0.77	0.9699	-0.18	402.66	116	142	REJECT
C15L4	0.41	0.9961	-0.20	246.40	238	275	**ACCEPT**
C15L5	0.48	0.9871	-0.43	227.71	153	183	REJECT
C15L6	0.53	0.9744	-0.72	124.83	92	115	REJECT
CUL4	0.32	0.9974	-0.41	255.37	229	265	**ACCEPT**
CUL5	0.22	0.9893	-0.51	205.34	183	216	**ACCEPT**
CUL6	0.28	0.9703	-0.69	126.78	86	109	REJECT

[59] Data concerning the unsuccessful model fittings can be found in Appendix A.

Table 6-17. C^2 values and best-fitting distributions

Representation	Zipf	MZ	GW	GIGP	Best Fit
C7L4 (VIEW A)	1126.90	1674.22	1674.22	*262.87*	**GIGP**
C7L4 (VIEW B)	1041.12	673.88	711.66	*307.89*	GIGP
C7L5 (VIEW A)	4107.91	1625.68	685.86	*387.94*	GIGP
C7L5 (VIEW B)	3418.74	857.02	748.07	*300.40*	GIGP
C7L6 (VIEW A)	9495.99	452.79	*216.29*	829.76	GW
C7L6 (VIEW B)	7208.49	447.42	*194.02*	402.66	GW
C15L4 (VIEW A)	1602.24	724.19	588.32	*318.16*	**GIGP**
C15L4 (VIEW B)	1147.69	699.07	492.30	*246.40*	**GIGP**
C15L5 (VIEW A)	3626.29	623.47	*404.73*	500.89	GW
C15L5 (VIEW B)	2076.52	650.99	312.46	*227.71*	GIGP
C15L6 (VIEW A)	4985.69	355.80	450.25	*123.59*	**GIGP**
C15L6 (VIEW B)	1951.92	435.33	185.93	*124.83*	GIGP
CUL4 (VIEW A)	851.05	555.95	420.52	*294.17*	**GIGP**
CUL4 (VIEW B)	558.19	774.93	425.51	*255.37*	**GIGP**
CUL5 (VIEW A)	1762.88	563.95	453.29	*370.59*	GIGP
CUL5 (VIEW B)	781.25	669.97	270.39	*205.34*	**GIGP**
CUL6 (VIEW A)	2998.97	404.48	634.44	*114.44*	**GIGP**
CUL6 (VIEW B)	810.09	488.35	198.27	*126.78*	GIGP
BOLDED = ACCEPT (p = 0.05) *ITALIC* = Lowest C^2					

That we have as many accepted fits as we do is quite remarkable given the very large degrees of freedom used in the goodness-of-fit tests.[60] However, where these data are most informative lies in reference to the findings of Nelson (1988). Nelson successfully fit the GIGP to the distribution of index terms found in the CACM and MEDLARS test collections. He also found the GW distribution to be the next-best fit for both (with the fit statistically significant for the CACM collection). One thing that the MEDLARS (1030 documents) and CACM (3204 documents) collections have in common is the relatively low frequency of their most frequently occurring types. In the CACM case, the most frequently occurring type has a document frequency of 46. In the MEDLARS case, there are only 10 types with a document frequency greater than 73. We

[60] Burrell and Fenton (1993) called "impressive" an accepted goodness-of-fit test with only 38 degrees of freedom. The degrees of freedom for our accepted models range from a minimum of 108 d.f. (C15L6 (View A)) to a maximum of 283 d.f. (*C15L4* (View A)).

have noted a similarly low maximum-occurrence rate for our *C7, C15*, and *CU* music databases (i.e., the absence of high-frequency music "stopwords")(see Table 6-11).

The fitting of the GIGP distribution is the third, and perhaps most important, characteristic that our databases and the CACM collection appear to have in common. The other two common characteristics are the mean (per record) tokens/type (Table 6-9) and the absence of high-frequency terms. Intrigued by these similarities, we decided to examine the CACM collection more closely. We discovered that the content fields of the CACM database generally consist of long titles, and, when present, rather short abstracts and/or short keyword lists. To illustrate this point, Figure 6-7 presents three randomly drawn records from the CACM collection. This discovery supports our earlier assertion that the n-grammed representations of the individual songs in our databases could be considered equivalent to short abstracts, and/or long scientific titles, and/or short keyword lists found in a text database.

```
.I [DocumentNumber] 20
.T [Title]
Accelerating Convergence of Iterative Processes
.W [Abstract]
A technique is discussed which, when applied to an iterative
procedure for the solution of an equation, accelerates the rate
of convergence if the iteration converges and induces convergence
if the iteration diverges.  An illustrative example is given.

.I [DocumentNumber] 30
.T [Title]
Algorithm for Analyzing Logical Statements to Produce a Truth
Function Table

.I [DocumentNumber] 1971
.T [Title]
Recorded Magnetic Tape for Information Interchange (1600CPI,
Phase Encoded)* (Proposed American National Standard)
.K [Keyword]
input-output, magnetic tape, information interchange,
measurement, instrumentation, phase encoded recording
```

Figure 6-7. Three sample records (abridged) from the CACM test collection

Nelson performed his informetric modeling upon the keyword fields of CACM and MEDLARS, so the absence of high-frequency terms is not surprising. While no such keyword field exists in our music databases, we note from their informetric similarity to the CACM and MEDLARS data that our databases also appear to be naturally "stopword-free." This fact, in conjunction with the other similarities evident between our

databases and the CACM collection, lead us to propose another equivalency that might be useful in understanding the potential retrieval characteristics of our music databases. We propose that one should consider the "full-text" searching and retrieval of records from our folksong databases ***not*** as equivalent to "full-text" searching of a text database (e.g., Altavista, InfoGlobe, etc.) but as the analogue of searching a "document surrogate" database (e.g., an OPAC, Library Literature, etc.) where the records, and the indexes that point to them, are heavily preprocessed, so that retrieval is performed only via a "concentrated" view of the information contained within the original documents. This equivalency implies retrieval performances for the music databases that will tend to emphasize precision over recall because document surrogates, unlike the documents for which they stand, tend to highlight, or concentrate, only the most significant of the key terms that describe the content of the documents (i.e., high term specificity, but low indexing exhaustivity). The concentration of key terms, along with the general brevity of the records, makes it less likely that a term used in a sought-after document would appear in an undesired document. Precision is thus enhanced because it is less likely that a given term would return an undesired document. Recall is hindered, however, because there are so few terms linking together the different records (i.e., most terms have very low document frequencies). While this equivalency has merit for understanding the potential retrieval characteristics of out folksong databases, it remains to be seen, however, what kind of equivalency would hold for music of different genres, and greater lengths.

Two important consequences flow from the examination of the similarities between the CACM collection and our music databases. First, it provides further support for our general hypothesis that there is sufficient information contained within the interval-only representation of monophonic melodies such that the n-gramming of interval-only melodic strings into "musical words" and their subsequent indexing will allow the same access to melodic information that indexes of "real words" give to textual information. Why is this so? Given that our music databases and the CACM collection have similar informetric characteristics, and given that other researchers have successfully used the CACM test collection numerous times within various text IR systems, it follows that our music databases should also be successful under a text IR

system.[61] Second, for want of a better text database analogue, we decided that we would use the CACM collection as the starting-point model to determine which type of term-weighting scheme to use in our Phase II retrieval evaluations. More about this follows next.

6.4 Implications for Phase II retrieval evaluations

As mentioned before, the first implication that these informetric analyses had on our Phase II evaluations was the dropping of the *C3* databases from further consideration. The second implication was the selection of a term-weighting scheme for use with our databases and the SMART retrieval system. The selection of a term-weighting scheme was based on the recommendations made by Salton and Buckley (1988) in their seminal overview, *Term-weighting approaches to automatic text retrieval.* In this study, the authors review the experimental results of the previous twenty years. Using the well-known CACM, CISI, CRAN, MED, and NPL collections, they go on to experimentally evaluate 1800 combinations of term-weighting schemes, find 287 distinct combinations, and then rank the 287 from best to worst. From these evaluations, Salton and Buckley derive detailed recommendations concerning the optimal term-weighting schemes to be used for various collection types.

Salton and Buckley (1988) represent the term-weighting combinations as a sextuple of the form:

Document vector		*Query vector*
$(TFC_d)\ (CFC_d)\ (NC_d)$	*	$(TFC_q)\ (CFC_q)\ (NC_q)$

where TFC = Term Frequency Component; CFC = Collection Frequency Component; NC = Normalization Component; and, *d* and *q*, signify document and query, respectively.

The TFC weights terms according to their within-record frequency (i.e., *tf*). However, high-frequency terms that are also widely distributed throughout a collection

---

[61] This argument, of course, does not *prove* that the retrieval performances of our music databases will be acceptable. However, it does show, based upon the aforementioned informetric similarities, that there is sufficient *prima facie* evidence to warrant the empirical evaluation of the retrieval performances of the *C7, C15,* and *CU* databases.

are poor discriminators. To compensate for this shortcoming, the CFC can be used to modify term weights. The CFC is a collection-dependent weighting factor that assigns greater weight to those terms that are concentrated in a few documents. A commonly used CFC is Sparck-Jones' (1972) *inverse document frequency* (IDF) factor. This is the same IDF factor used in Table 6-7. Variations in both document and query vector lengths can adversely affect retrieval effectiveness so the NC can be added to equalize the vector lengths when necessary.

For the CACM collection Salton and Buckley (1988) report the best results with a document term-weighting scheme of:

TFCd = tf, or raw term frequency

CFC_d = IDF, or $\log_2 N/n$, where *N* is the number of documents in the collection, and *n* the number of documents in which the term occurs

$NC_d = \frac{1}{\sqrt{\sum_{vector} w_i^2}}$ which is the cosine normalization where each term weight *w* is divided by a factor representing the Euclidean vector length

However, they also note that the NPL collection, with its low mean (per record) tokens/type (i.e., *tf*) value required the use of a TFC_d which boosted the effect of the *tf* values. For shorter queries, a TFC_q which boosted the effect of a query's *tf* value also helped performance. The TFC that they recommend in such situations is called the "augmented normalized term frequency." Since our music databases have *tf* values closer to the NPL collection's than the CACM's, and since our queries will be short by text standards, we decided to substitute the augmented normalized term frequency for the normal *tf* weight. Thus, our final choice of term-weighting became:

$TFC_{(d \text{ and } q)} = 0.5 + 0.5 \frac{tf}{tf_{max}}$

$CFC_{(d \text{ and } q)} = \log_2 N/n$

$NC_{(d \text{ and } q)} = \frac{1}{\sqrt{\sum_{vector} w_i^2}}$

6.5 Summary

In this chapter we have reported upon the Phase I informetric analyses undertaken to comprehend the potential utility of the MusiFind approach to MIR. The remarkable consistency in the data, given the wide variety of analyses performed, led us to the following conclusions and actions.

We have eliminated from further investigation the *C3* databases because of their consistently inferior informetric properties. We have retained the *C7* databases principally because our analyses of their informetric properties were ambiguous with regard to potential retrieval effectiveness. The *C15* and *CU* databases were also retained because of their consistently superior informetric characteristics that suggest strong retrieval performances. We also noted that the slight differences between the *C15* and *CU* databases suggest only marginal, if any, difference in retrieval success between these sets of databases.

We have rejected the hypothesized equivalency between intervals and letters, but have retained, for its pragmatic and metaphoric utility, the hypothesized equivalency between n-grams and words (Subsidiary hypothesis 1). We have proposed a new equivalency that might assist in better understanding the retrieval characteristics of our music databases. The CACM collection, which is a document surrogate database, is the strongest analogue we have so far encountered to our music databases. Thus, for the purposes of retrieval evaluation, our music databases are best thought of as equivalent to textual document surrogate databases rather than full-text databases. We have used this equivalency as the starting-point model for determining our choice of term-weighting scheme vis-à-vis Salton and Buckley's (1988) term-weighting recommendations.

With the exception of the *C3* databases, we have uncovered no evidence that suggests that the MusiFind approach to MIR will not provide the same access to melodic information that text IR systems provide for text (Principal hypothesis). To the contrary, our analyses of the informetric data suggest that the retrieval performances of our music databases under the SMART system should range from good (*C7L4*) to excellent (*C15L6* and *CUL6*).

7 Phase II: Information Retrieval Evaluations

7.1 Introduction[62]

In this chapter, we report upon our experimental IR simulations and evaluations. We will begin with a delineation of our experimental design, our sampling methods, our query and error simulation procedures, our formal hypothesis concerning main effects, and the statistical methods employed to evaluate our results. We will present our results and highlight their salient features, first descriptively and then in light of the tests for statistical significance.

As one will see, the tests for statistical significance indicate a complex interaction at play between four of our five independent factors CLASS, NLEN, QLEN, QLOC, and QQUAL.[63] Considerable time will be spent simplifying and explicating these complex interactions in order to make meaningful and cogent design recommendations for MIR system development. We will revisit our informal questions concerning the independent factors (see Chapter 5), and through this re-visitation we will present and justify our design recommendations. Before concluding this chapter we will also revisit the document space density *Q* data, calculated during the Phase I informetric analyses, to develop a model of MIR system performance grounded in the indexing theory of Salton (1975). We will conclude this chapter by presenting our decisions concerning our principal and subsidiary hypotheses.

7.2 Methods

7.2.1 Analytic tools

Three sets of analytic tools were used in our evaluations. The first set was a series of PERL programmes written by the author (e.g., query selection, query processing, error simulation, data collection, data conversion, etc.). The retrieval reporting sub-routines integral to the SMART system were the second set. The third set were the statistical processing functions of SPSS for Windows 8.0.0 (1997).

---

[62] This chapter represents a comprehensive revision of Downie 1999a and 1999b.

[63] Again, we continue to use the study nomenclature presented in Chapter 5.

The SMART retrieval system was used "as-is" with the exception of one slight modification. The default procedure for SMART is to ignore case. However, our melodic representations use case to signify interval direction so the SMART source code was changed to disable its case-conflation procedures. We believe that this slight modification in no way deviates from our stated goal of using a text retrieval system "off-the-shelf" (Paradigm 1).

7.2.2 Experimental design

Melodic strings, extracted from the songs in our Baseline Database (BD) file, were used as our queries. The extraction of strings from database records for use as experimental queries is noted by Tague-Sutcliffe (1992) as a historically valid method. The experimental model was a complex, mixed-, five-way factorial design. Retrieval effectiveness was evaluated under 108 experimental conditions as Table 7-1 illustrates. In total, 3240 query runs were made.

Table 7-1. Experimental factors

CLASS	NLEN	QLEN	QLOC	QQUAL	Total
Within	Within	Within	Between	Within	
C7, C15, CU	L4, L5, L6	Q6, Q8, Q10	I, R	P, E	
3 **X**	3 **X**	3 **X**	2 **X**	2 **X**	= 108

In our design, the queries functioned as our experimental subjects (*Ss*). The *Ss* were crossed with the levels of the CLASS, NLEN, QLEN and QQUAL factors. The *Ss* were also nested under the two levels of the QLOC factor. Tague-Sutcliffe (1992) notes that queries have a strong tendency to introduce a great deal of variance. She adds that this variance must be controlled through the use of some type of repeated-measures design, otherwise the treatment effects can be masked by the unwanted *Ss* variance. Therefore, our experimental design relied heavily upon within-subject measurement in order to minimize the effect of the variance introduced by the experimental *Ss* (i.e., the queries).

Table 7-2. Experimental design

Cell *N* = 30			QLOC: *I*ncipit						*R*andom					
			QQUAL: *P*erfect			*E*rror			*P*erfect			*E*rror		
			CLASS			CLASS			CLASS			CLASS		
			7	15	CU	7	15	CU	7	15	CU	7	15	CU
QLEN	6	NLEN 4												
		NLEN 5												
		NLEN 6												
	8	NLEN 4												
		NLEN 5												
		NLEN 6												
	10	NLEN 4												
		NLEN 5												
		NLEN 6												

7.2.3 Formal hypothesis concerning main effects

$$H_0 : m_{CLASS} = m_{NLEN} = m_{QLEN} = m_{QLOC} = m_{QQUAL}$$

$$H_A : m_{CLASS} \neq m_{NLEN} \neq m_{QLEN} \neq m_{QLOC} \neq m_{QQUAL}$$

Simply put, our null hypothesis was that there would be no difference in retrieval effectiveness caused by the experimental factors. Our alternative hypothesis was that the experimental factors would affect retrieval performance. Also, specific *a priori* tests for differences between factor levels were conducted as explained below in Chapter 7.2.8.

7.2.4 Sampling method

A PERL programme was written by us to generate random numbers ranging from 1 through 9354, inclusive (i.e., the number of songs in our databases). After thirty unique numbers had been generated, copies of those songs with accession numbers corresponding to the randomly generated numbers were taken from the BD file. A sub-

string of length-11 was then extracted from the beginning of each selected song.[64] These thirty strings became the "raw" versions of the *Incipit* queries that were nested under level *I* of the QLOC factor, and crossed with all others. The strings were also copied into a file called the *Incipit* Query Source (*IQS*) file for use in the query creation process.

After initializing our random-number programme with a new seed, we generated another set of thirty numbers.[65] The corresponding songs were copied into a temporary holding file. A second random-number programme was then called to generate a randomly selected location within each song from which the length-11 sub-strings could be extracted. The second programme generated random numbers under two constraints. First, the location had to be 11 intervals from the song end to ensure a length-11 query string. Second, the selected location could not represent a song's first interval as this would merely replicate an *Incipit* query. The thirty sub-strings thus extracted represented the "raw" forms of the Random queries that were nested under level *R* of QLOC, and crossed with all others. The resultant strings were copied into the Random Query Source (*RQS*) file for use in the query creation process.

7.2.5 Query creation process

The simplest way to explain the query creation process is through a series of examples.

> **Example 1**: Imagine that the *C7L6IQ7P* condition was examined. That is to say, thirty perfect (*P*) incipit (*I*) queries of length-7 (*Q7*) were submitted as queries against the database where the melodies are represented by n-grams of length-6 (*L6*), comprising intervals classified into seven categories (*C7*). From the beginning of each of the thirty strings located in the *IQS* file was extracted a single sub-string of length-7. These

---

[64] The maximum QLEN to be evaluated was *Q10*. However, the extra interval was extracted as a matter of convenience. The extra interval was "on standby" should it have been required to restore QLEN to *Q10* after a deletion operation used in the simulation of query errors (QQUAL). The error-simulation process is discussed in Chapter 7.2.6.

[65] There was no duplication in the numbers generated between the first and second runs.

contiguous, unclassified sub-strings represent what we came to call the "query progenitor strings."[66] Each of these progenitor strings was translated into its *C7* representation. Next, each of these newly classified strings was parsed into n-grams of the appropriate length, which, in this example case, were two n-grams of length-6. This completed the query creation process. All of the queries used in the *Incipit* by Perfect (*I* by *P*) cells were created in this manner.

Example 2: Imagine that the *C7L6IQ7E* condition was examined. The procedure replicated Example 1 except that each of the queries was modified to contain a simulated error (*E*). Again, to create the query string progenitor, sub-strings of length-6 were extracted from the beginning of each of the thirty strings located in the *IQS* file. Before undergoing the *C7* classification process, however, each of the sub-strings underwent the error simulation process described in the next section. Once classified, the n-gramming process was applied as above. All of the queries used in the *Incipit* by Error (*I* by *E*) cells were created in this manner.

Example 3: Imagine that the *C7L6RQ7P* condition was examined. The only difference between this example and Example 1 is that the level of the QLOC factor has been changed from *I* (*Incipit*) to *R* (Random). The procedure given in Example 1 was followed except that the progenitor strings were taken from the *RQS* file rather than the *IQS* file. All of the queries used in the Random by Perfect (*R* by *P*) cells were created in this manner.

Example 4: Finally, imagine that the *C7L6RQ7E* condition was examined. The procedure given in Example 2 was followed except the

---

[66] There is a length-6, -8, and -10 progenitor string for each of the *Q6*, *Q8*, and *Q10* queries, respectively. A progenitor string represent a query in its "raw," unclassified, and non-n-grammed form.

progenitor strings were taken from the *RQS* file. All of the queries used in the Random by Error (*R* by *E*) cells were created in this manner.

7.2.6 Error simulation procedure

A PERL programme, written by us, completed the tasks necessary to simulate user errors. The error simulations were performed by taking each progenitor string and randomly selecting one interval *n* at position *y* where *y* varied from 1 to *x* (i.e., the end of the string). Once selected, the interval type of *n* was then identified. The specific modification made to the progenitor interval was determined randomly according to rules appropriate for the type of interval selected. The appropriate rules for each interval type are found in Table 7-3. These rules served as our best approximation of the user errors noted by McNab et al. (1996) and discussed in Chapter 5.

Table 7-3. Error simulation rules

Original *n*	Interval Range	Error Type	Replace *n* with one of the following at probability *p*	Probability factor *p*
1 <= *n* <= 4	Small Positive	Expansion	*n* + 1	0.40
		Repetition	*n* = 0	0.40
		Omission	delete *n*	0.20
-1 >= *n* => -4	Small Negative	Expansion	*n* – 1	0.40
		Repetition	*n* = 0	0.40
		Omission	delete *n*	0.20
n > = 5	Large Positive	Compression	*n* – 1	0.40
		Repetition	*n* = 0	0.40
		Omission	delete *n*	0.20
n < =- 5	Large Negative	Compression	*n* + 1	0.40
		Repetition	*n* = 0	0.40
		Omission	delete *n*	0.20
n = 0	Zero	Sharpening	*n* + 1	0.30
		Flattening	n – 1	0.30
		Repetition	insert another 0	0.30
		Omission	delete n	0.10

It was important for reasons of statistical validity that the lengths of the modified query stings remained consistent with their unmodified versions (i.e., the progenitor strings). If the decision to delete an original interval *n* was made, the intervals at positions *y* –1 and *y* +1 were conjoined yielding a string *x* –1 in length. To restore the QLEN *Q*(*x* –1) to *Qx* (i.e., x = 6, 8, or 10), the interval at position *x* +1 in the source

melodic string was added to the end of the modified query to create the new query string *QxE*. This is why we stored the length-11 strings in our *IQS* and *RQS* files.

The **Interval Range: Zero** condition (Table 7-3) requires special explanation. McNab et al. (1996) are unclear about what types of errors occurred when the original $n = 0$. While it is illogical to simulate a *Compression* error when an interval n equals 0, it is not unlikely to conceive of situations where a user, singing a query *a cappella*, might suffer from vocal fatigue or stress. Such fatigue or stress might cause the pitch of the query to drop or rise enough that a pitch- tracking interface might register the next iteration of the pitch as a different note, 1 semitone. Thus, when an interval n fell within the Interval Range: Zero condition, a *Sharpening* or *Flattening* error was simulated by replacing $n = 0$ with $n + 1$ or $n - 1$, respectively. In fact, these error types are special instances of the Expansion error type.

7.2.7 Dependent measures

The dependent measures were the normalized precision (NPREC) and normalized recall (NREC) metrics discussed in Chapter 5. The set of relevant documents for a given query was defined as being the set of those songs in which the query's progenitor string was found intact. Of these two measures, the NPREC is taken by us to be the more important. Tonta (1992) cites a growing body of literature that contends that users of IR systems are much more satisfied with search sessions that have strong precision rather than strong recall. Therefore, we decided, notwithstanding our interest in the recall performance of our approach, that we would focus our attention upon its precision performance. Thus, our forthcoming analyses, discussions, conclusions and recommendations, will be based solely upon the NPREC data.

7.2.8 Analytic methods

Data were analyzed under the Repeated Measures General Linear Model functions of SPSS (1997). The test statistics were calculated according to the multivariate analysis of variance (MANOVA) approach to repeated measures. The MANOVA approach is particularly well suited for complex experimental designs, as it is a member of the analysis of variance (ANOVA) "family" of statistical procedures. Both Kinnucan, Nelson and Allen (1987), and Tague-Sutcliffe (1992), recommend the

ANOVA family of procedures as the methods of choice for comprehensive IR evaluations. ANOVA-style evaluations do not fall prey to the compounding of Type I error that use of multiple *t*-tests incurs. There is an univariate approach to repeated measures that is more powerful, *ceteris paribus*, than the MANOVA approach; however, it has strict assumptions concerning the homogeneity of error variance, assumptions that are rarely met (Marascuilo and Levin 1983). Because of the method of error partitioning employed by the MANOVA approach, it is a very robust analytic method with regard to assumptions of population normality and homogeneity of error-variance. With equal cell counts and large *N* —conditions met by our design— the MANOVA approach to within-subject testing is considered to be a highly reliable analytic tool (Marascuilo and Levin 1987).

To increase the power of the MANOVA the retrieval data were subjected to an ArcSin(Sqrt(x)) transformation prior to analysis.[67] Tague-Sutcliffe (1992) specifically recommends this type of data transformation for use with precision and recall data. Because recall and precision data are bounded between 0 and 1, assumptions of normality are difficult to meet. Fundamentally, the ArcSin(Sqrt(x)) transformation expands the range of the data beyond 1which in turn makes them more normally distributed.

In addition to the omnibus MANOVA tests for main effects, REPEATED (SPSS 1997) contrast codes were employed. These contrast codes represented *a priori* tests for differences between the levels of the within-subjects factors CLASS, NLEN, QLEN and QQUAL. More precisely, REPEATED contrast codes "compare the mean of each level (except the last) to the mean of the subsequent level" (SPSS 1997). Because these were *a priori*, single degree-of-freedom tests, they were both conservative and did not increase the Type I error rate as *post hoc* multiple means comparisons would have.

---

[67] That is, the arcsine of the square root of each datum was taken as the transformed value.

7.3 Data and analyses

7.3.1 Descriptive data

The results of the retrieval runs are presented below. We have summarized these data with an eye toward their usefulness in making design recommendations. Therefore, the data presented in Table 7-4 (NPREC) and Table 7-5 (NREC) represent the results averaged over all levels of QLEN and both levels of QLOC. We have averaged over all levels of QLEN because QLEN represents a "user factor" over which a system designer would have little or no control. We have averaged over both levels of QLOC because in neither the NPREC nor NREC tests were the results statistically significant. Table 7-6 (NPREC) and Table 7-7 (NREC) present the results of the between-subjects tests. Figure 7-1 (NPREC) and Figure 7-5 (NREC) present the QLOC data in graphic form. The QLEN data are charted in Figure 7-4 (NPREC) and Figure 7-8 (NREC).

Table 7-4. NPREC descriptive summaries (averaged over queries of all lengths (QLEN) and both locations (QLOC))

	NLEN											
	L4			L5			L6			AVE (CLASS)		
	QQUAL			QQUAL			QQUAL			QQUAL		
CLASS	P	E	AVE (P&E)	P	E	AVE (P&E)	P	E	AVE (P&E)	P	E	AVE (P&E)
C7	.8874	.6319	.7596	.9330	.6109	.7720	.9485	.5114	.7300	.9230	.5847	.7539
C15	.9438	.6909	.8174	.9781	.6481	.8131	.9905	.4883	.7394	.9708	.6091	.7900
CU	.9445	.6952	.8199	.9807	.6489	.8148	.9927	.4869	.7398	.9727	.6103	.7915
AVE (NLEN)	.9252	.6727	.7990	.9639	.6360	.8000	.9773	.4955	.7364	.9555	.6014	.7785

Table 7-5. NREC descriptive summaries (averaged over queries of all lengths (QLEN) and both locations (QLOC))

	NLEN											
	L4			L5			L6			AVE (CLASS)		
	QQUAL			QQUAL			QQUAL			QQUAL		
CLASS	P	E	AVE (P&E)	P	E	AVE (P&E)	P	E	AVE (P&E)	P	E	AVE (P&E)
C7	.9996	.9751	.9874	.9999	.9417	.9708	.9999	.8537	.9268	.9998	.9235	.9617
C15	.9999	.9627	.9813	1.0000	.9143	.9572	1.0000	.7825	.8913	.9999	.8865	.9432
CU	.9999	.9626	.9812	1.0000	.9047	.9523	1.0000	.7796	.8898	.9999	.8823	.9411
AVE (NLEN)	.9998	.9668	.9833	.9999	.9202	.9601	1.0000	.8053	.9026	.9999	.8974	.9487

Table 7-6. NPREC test of between-subjects effects (QLOC)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Eta Squared
Intercept	84.923	1	84.923	4741.165	0	0.988
QLOC	2.12E-04	1	2.12E-04	0.012	0.914	0
Error	1.039	58	1.79E-02			

Table 7-7. NREC test of between-subjects effects (QLOC)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Eta Squared
Intercept	128.985	1	128.985	26210.88	0	0.998
QLOC	4.25E-03	1	4.25E-03	0.863	0.357	0.015
Error	0.285	58	4.92E-03			

With regard to normalized precision (NPREC) (Table 7-4; Figure 7-1 through Figure 7-4) the overall results are quite positive. The nominally best performance comes from the *CUL6P* condition (Unclassified intervals, Length-6 n-gram, Perfect query) which has a NPREC (averaged over all query lengths) of 0.9927. The nominally best performance under the *Error* condition is *CUL4E* (0.6952). *CUL4* also returns the best performance when the *Perfect* and *Error* conditions are averaged (0.8199).

As one looks through the charts presented in Figure 7-1 through Figure 7-8 four highlights are worth noting. First, CLASS (Figure 7-2) and NLEN (Figure 7-3) appear to be partitioned as the informetric data suggested they would (i.e., *C7* < (*C15* = *CU*) and *L4* < *L5* < *L6*). Second, NLEN appears to interact with QQUAL. This interaction is manifest by the divergent QQUAL curves as one progresses through the NLEN levels. Third, the QQUAL curves converge as one progresses through the QLEN (Figure 7-4) levels, which suggests a QLEN by QQUAL interaction. Fourth, the distance between the QQUAL *Perfect* and the QQUAL *Error* curves appears consistently large enough to represent a statistically significant result (Figures 7-1 through 7-4).

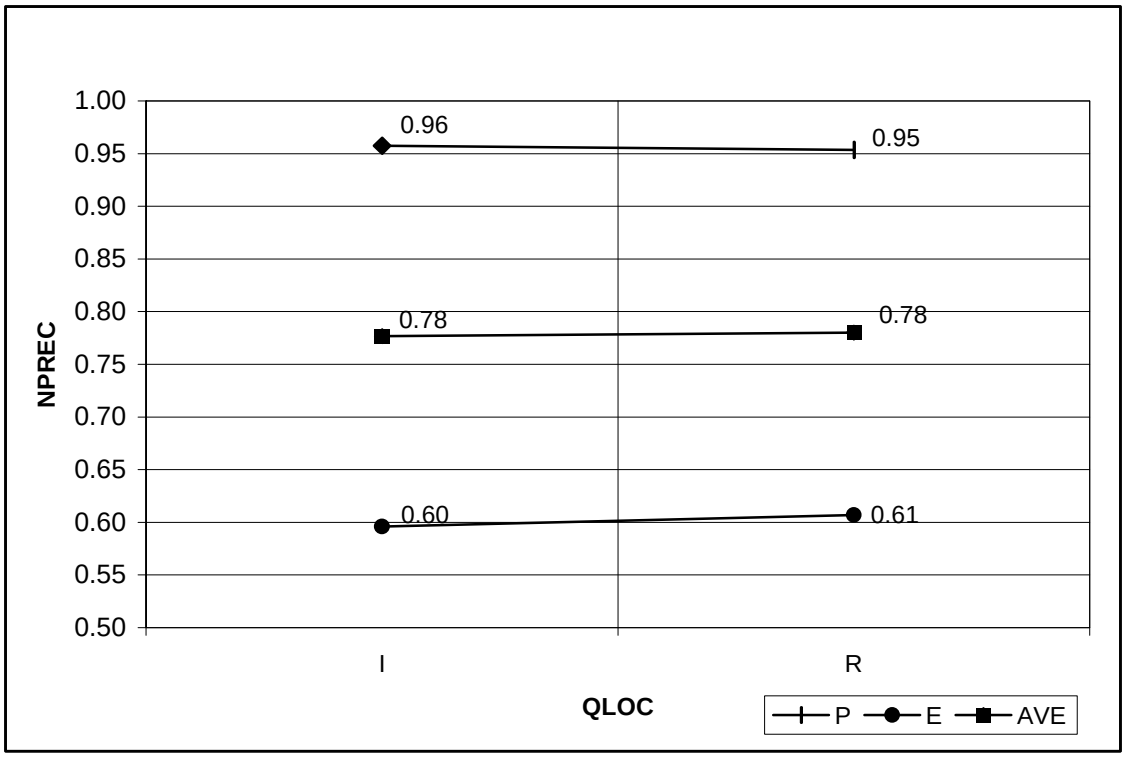


Figure 7-1. NPREC by QLOC

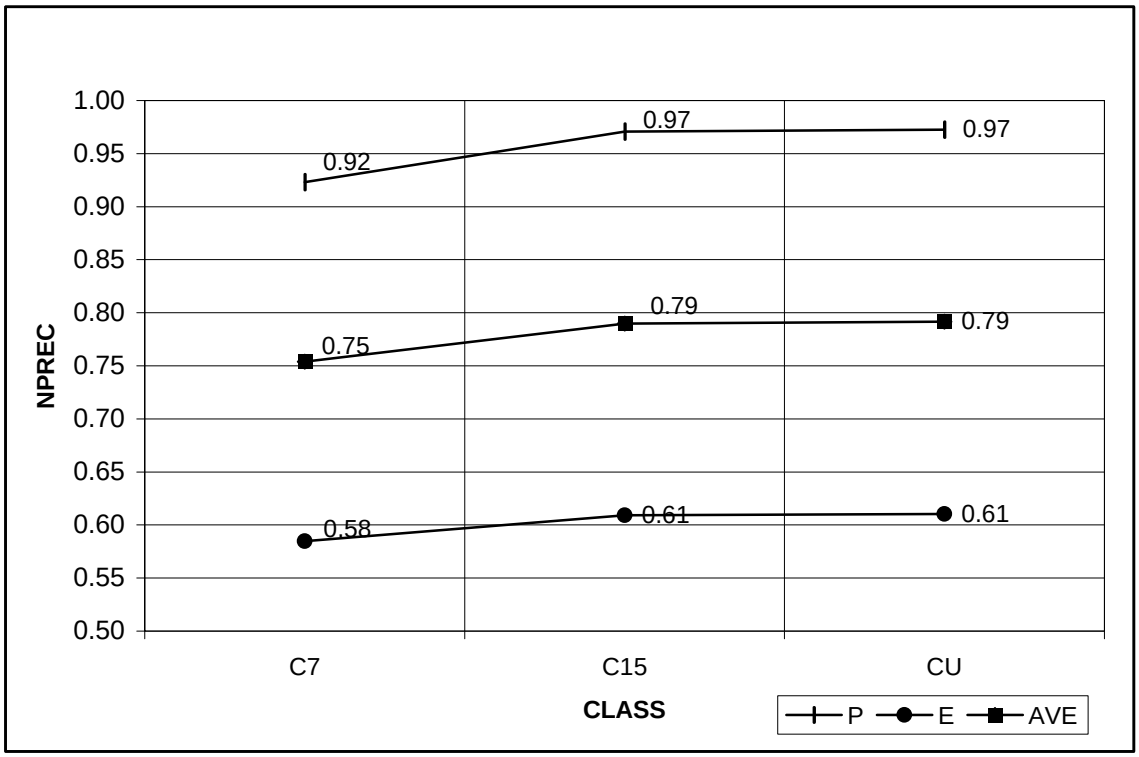


Figure 7-2. NPREC by CLASS

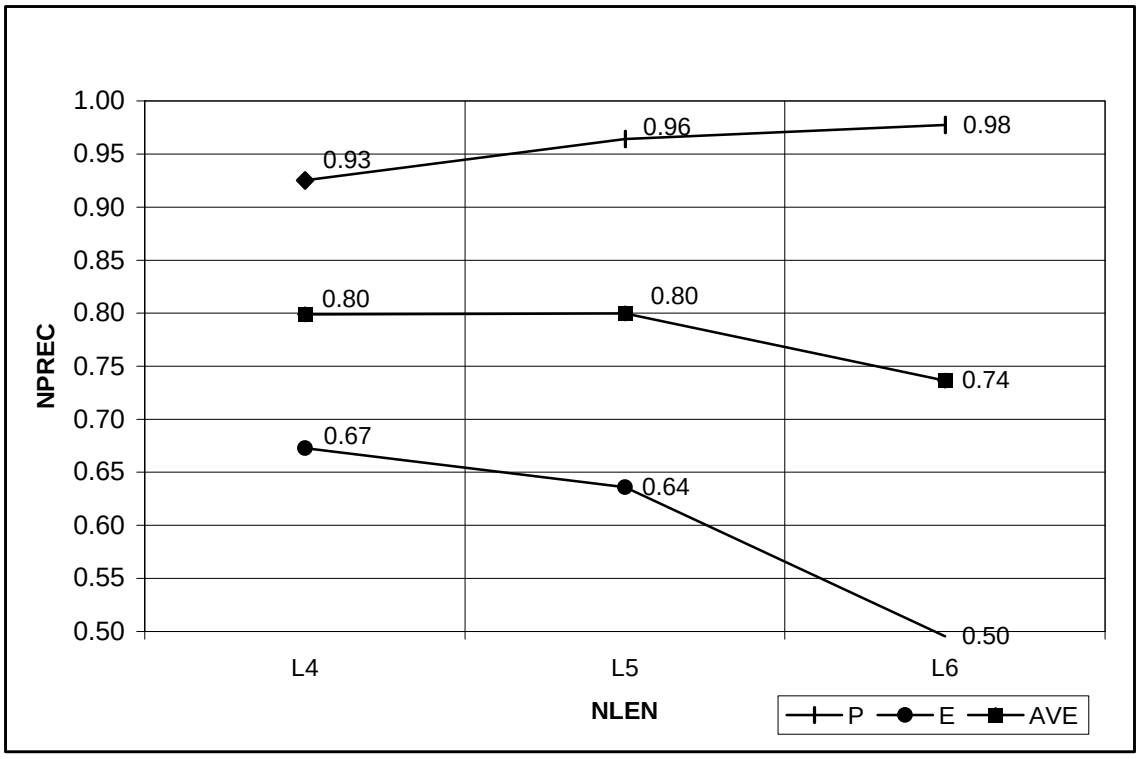


Figure 7-3. NPREC by NLEN

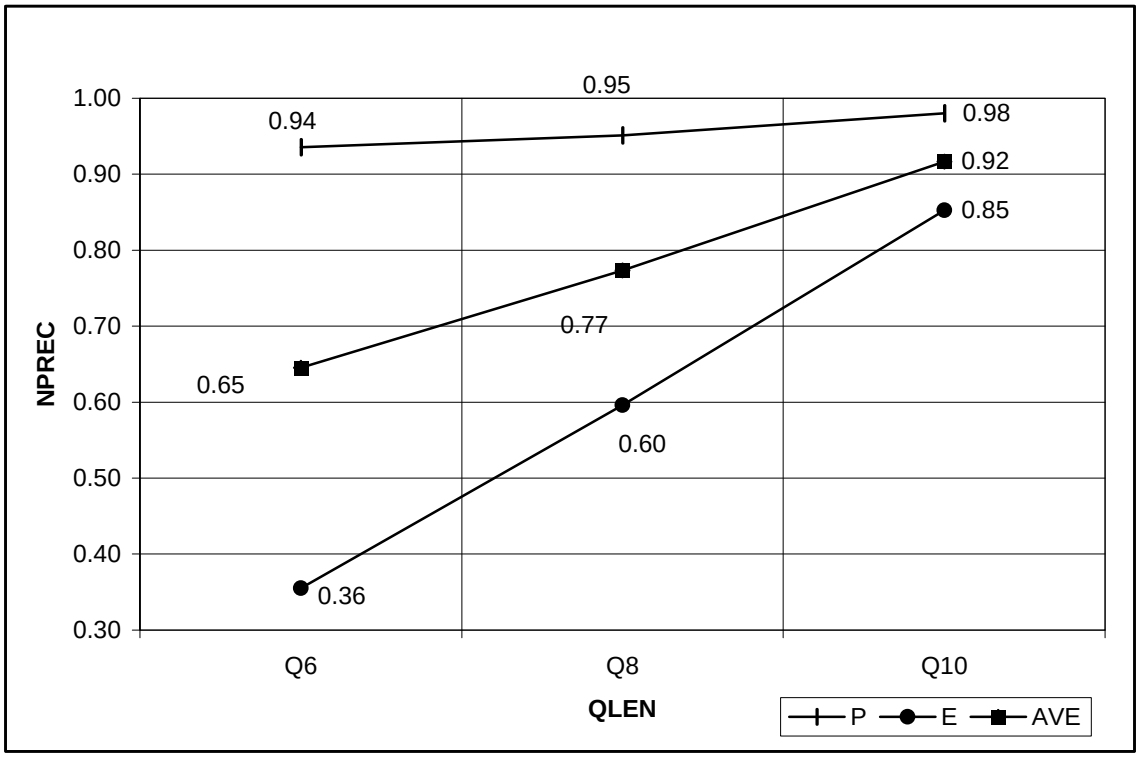


Figure 7-4. NPREC by QLEN

Figure 7-5. NREC by QLOC

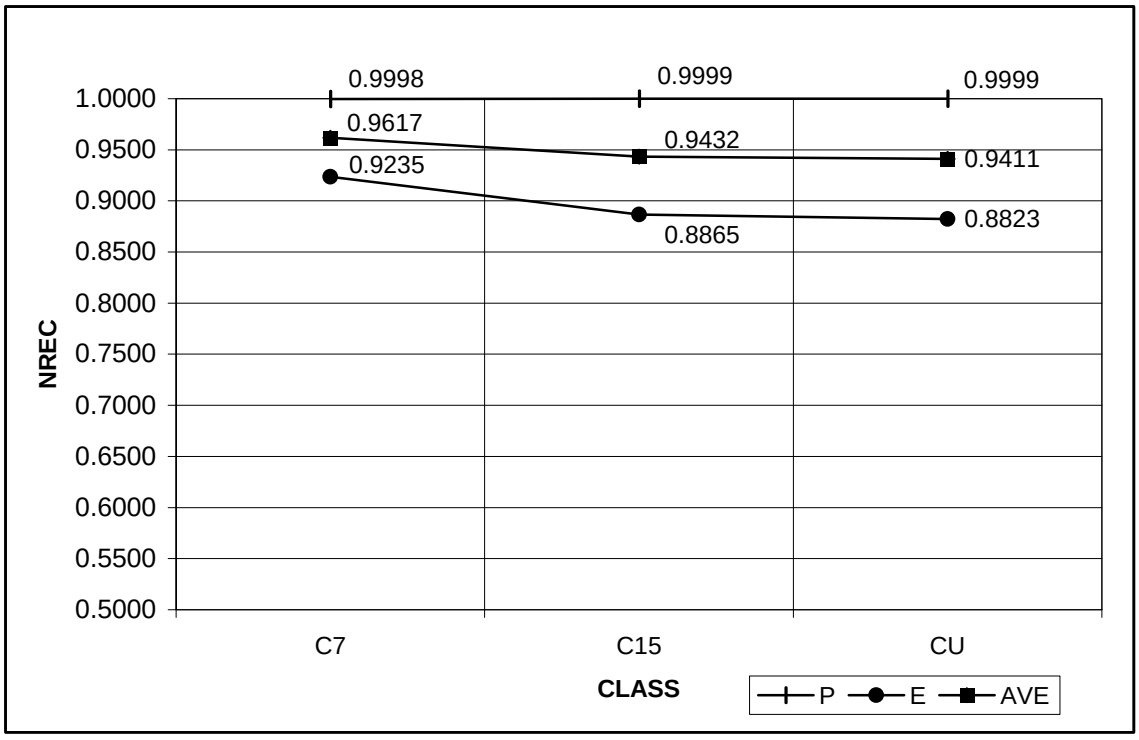


Figure 7-6. NREC by CLASS

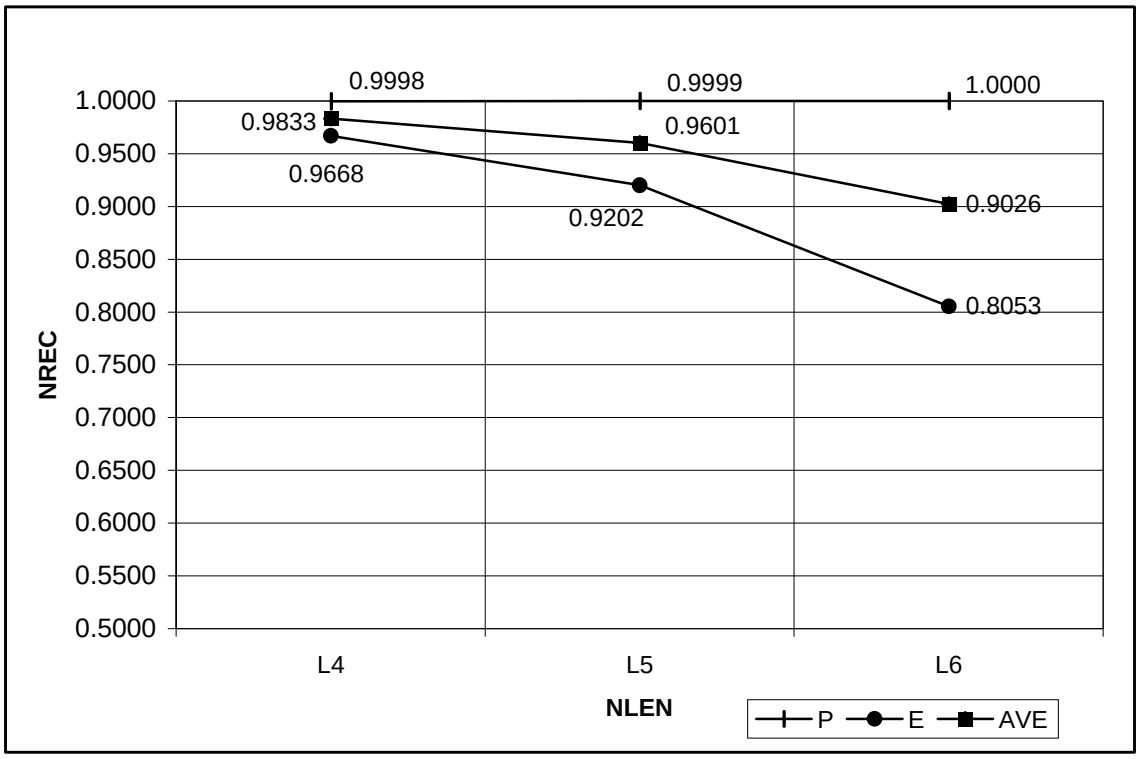


Figure 7-7. NREC by NLEN

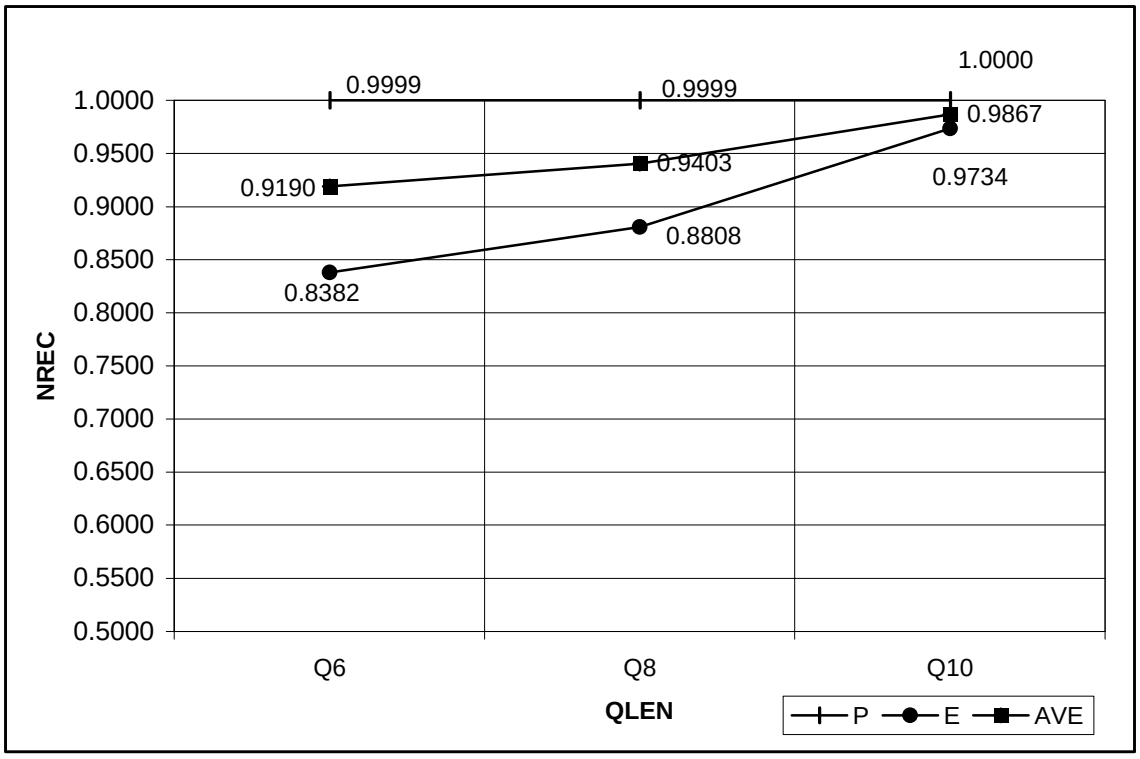


Figure 7-8. NREC by QLEN

7.3.2 Statistical analyses

Presented below are *select* results of the omnibus multivariate tests of within-subject effects (NPREC) and Table 7-9 (NREC)).[68] We have presented here only those statistically significant results ($p < 0.05$) that either represent main effects or have an Eta Squared (η^2) value greater than 0.20. We have done this primarily as a means of clarifying the presentation of our results. Eta Squared is a measure of the proportion of variance in the dependent variable accounted for by the independent variable(s) undergoing testing. As such, η^2 is an estimator of the size of the effect attributable to the tested variable(s). Eta Squared values range from 0 (i.e., the independent variable(s)

[68] Complete results of the multivariate tests of within-subject effects can be found in Appendix B.

create(s) no effect in the dependent variable) through to 1 (i.e., perfect correlation between independent and dependent variable(s)).[69]

Table 7-8. NPREC: Select multivariate tests of within-subject effects

Effect	F	df	Error df	Sig.	Eta Squared
CLASS	12.774	2	57	0	0.309
NLEN	16.571	2	57	0	0.368
QLEN	89.165	2	57	0	0.758
QQUAL	442.557	1	58	0	0.884
CLASS * NLEN	5.368	4	55	0.001	0.281
CLASS * QQUAL	10.658	2	57	0	0.272
NLEN * QLEN	3.445	4	55	0.014	0.200
NLEN * QQUAL	79.143	2	57	0	0.735
QLEN * QQUAL	53.788	2	57	0	0.654
CLASS * NLEN * QQUAL	13.606	4	55	0	0.497
CLASS * QLEN * QQUAL	14.396	4	55	0	0.511
NLEN * QLEN * QQUAL	20.294	4	55	0	0.596
CLASS * NLEN * QLEN * QQUAL	2.840	8	51	0.011	0.308

Table 7-9. NREC: Select multivariate tests of within-subject effects

Effect	F	df	Error df	Sig.	Eta Squared
CLASS	3.772	2	57	0.029	0.117
NLEN	36.102	2	57	0	0.559
QLEN	22.634	2	57	0	0.443
QQUAL	117.503	1	58	0	0.670
CLASS * NLEN	6.477	4	55	0	0.320
NLEN * QLEN	6.726	4	55	0	0.328
NLEN * QQUAL	41.384	2	57	0	0.592
QLEN * QQUAL	21.252	2	57	0	0.427
CLASS * NLEN * QQUAL	5.999	4	55	0	0.304
NLEN * QLEN * QQUAL	7.794	4	55	0	0.362

[69] Kinnucan, Nelson and Allen (1987) commend the use of effect-size estimation, though they note, at the time of their writing, that it was very rare to see such estimates in the information science literature. Our preparatory reading of the literature for this research project confirms their earlier finding in that we encountered no reports of effect-size estimation. We find this surprising given the prevalence of statistical packages that can provide the estimates with next-to-no effort on the part of the researcher. Furthermore, we have found the estimates of effect size to be a most useful tool in our analyses in that they helped us to go beyond mere statistical significance to uncover what we believe to be the truly more important relationships.

The multivariate tests of within-subject effects bring out two important facts. First, all the within-subject factors create statistically significant main effects. Based upon these results, we therefore reject our null hypothesis. In its place we accept that CLASS, NLEN, QLEN and QQUAL all influence retrieval performance. Second, and analytically more problematic, are the complex, statistically significant 2-, 3-, and 4-way interactions of the factors.

Meaningful interpretations of multi-way interactions are extraordinarily difficult to provide (Kinnucan, Nelson and Allen 1987). We believe, however, that we can provide some meaning to these interactions through careful analyses of the *Error* level of the QQUAL factor and its interaction with NLEN and QLEN. We will return to this matter presently, but for now note how the 3-, and 4-way interactions all involve QQUAL. Also, note how the QQUAL interactions consistently have the higher η^2 values.

The NPREC η^2 values (Table 7-8) allow us to rank-order our factors based upon their relative influence on retrieval performance. CLASS is the least-influential factor with an η^2 of 0.309. NLEN is next with $\eta^2 = 0.368$. QLEN is the second-most influential factor (0.758). With an η^2 value of 0.884, QQUAL is the most influential factor. We find it noteworthy that CLASS and NLEN, "system factors" under the control of designers, have the least amount of influence on retrieval performance while QLEN and QQUAL, both of which are "user factors," and thus generally outside of the control of system designers, have the greatest amount of influence. This finding suggests that further research into the study of actual human-system interaction would provide important and potentially beneficial data for the further refinement of any MIR system developed. The ultimate goal of such research would be the lowering of the η^2 values for QLEN and QQUAL to such an extent that these factors would cease to have any meaningful influence on retrieval performance.

The tests of within-subject contrasts presented in Table 7-10 (NPREC) and Table 7-11 (NREC) are very informative with regard to two important issues. First, they confirm our suspicions brought forth during the informetric analyses concerning the grouping of the CLASS results (i.e., (*CU =C15*) > C7). Second, the persistent

interaction of the user factors, QQUAL and QLEN, with various levels of the system factors, CLASS and NLEN, makes it even more apparent that no clear-cut determination of a "best" combination of system factors can be made. For example, *CUL6* has both the best, and the worst, NPREC results: *CUL6P* (0.9927) and *CUL6E* (0.4869). Notwithstanding the inherent complexities involved with interpreting the multi-way interactions, the NPREC tests of within-subject contrasts, along with the between-subject test (QLOC), can be summarized and brought together to show:

CLASS: *CU* (0.7915) and *C15* (0.7900) superior to *C7* (0.7539)
No significant difference *C15 – CU*

NLEN: *L5* (0.8000) and *L4* (0.7990) superior to *L6* (0.7364)
No significant difference *L4 – L5*

QLEN: *Q10* (0.9164) superior to *Q8* (0.7735)
Q8 (0.7735) superior to *Q6* (0.6453)

QQUAL: *P* (0.9554) superior to *E* (0.6014)

QLOC: No significant difference *I* (0.7767) – *R* (0.7801)

Table 7-10. NPREC: Significant tests of within-subject contrasts

Source	CLASS	NLEN	QLEN	QQUAL	df	F	Sig.	Eta Squared
CLASS	C7 vs. C15				1	24.800	0	0.300
NLEN		L5 vs. L6			1	30.856	0	0.347
QLEN			Q6 vs. Q8		1	21.893	0	0.274
QLEN			Q8 vs. Q10		1	55.440	0	0.489
QQUAL				P vs. E	1	442.557	0	0.884
CLASS * NLEN	C7 vs. C15	L5 vs. L6			1	18.747	0	0.244
CLASS * QQUAL	C7 vs. C15			P vs. E	1	18.021	0	0.237
NLEN * QLEN			Q8 vs. Q10		1	4.630	0.036	0.074
NLEN * QQUAL		L4 vs. L5		P vs. E	1	63.818	0	0.524
NLEN * QQUAL		L5 vs. L6		P vs. E	1	73.757	0	0.560
QLEN * QQUAL			Q6 vs. Q8	P vs. E	1	14.842	0	0.204
QLEN * QQUAL			Q8 vs. Q10	P vs. E	1	26.991	0	0.318
CLASS * NLEN * QLEN	C7 vs. C15	L4 vs. L5	Q6 vs. Q8		1	4.260	0.044	0.068
CLASS * NLEN * QLEN		L5 vs. L6	Q8 vs. Q10		1	5.311	0.025	0.084
CLASS * NLEN * QLEN		L5 vs. L6	Q8 vs. Q10		1	4.627	0.036	0.074
CLASS * NLEN * QQUAL	C7 vs. C15	L5 vs. L6		P vs. E	1	36.569	0	0.387
CLASS * NLEN * QQUAL	C7 vs. C15	L4 vs. L5		P vs. E	1	5.565	0.022	0.088
CLASS * QLEN * QQUAL	C7 vs. C15		Q6 vs. Q8	P vs. E	1	34.216	0	0.371
CLASS * QLEN * QQUAL	C7 vs. C15		Q8 vs.Q10	P vs. E	1	8.624	0.005	0.129
CLASS * QLEN * QQUAL	C15 vs. CU		Q6 vs. Q8	P vs. E	1	8.770	0.004	0.131
NLEN * QLEN * QQUAL		L4 vs. L5	Q6 vs. Q8	P vs. E	1	4.906	0.031	0.078
NLEN * QLEN * QQUAL		L4 vs. L5	Q8 vs. Q10	P vs. E	1	16.080	0	0.217
NLEN * QLEN * QQUAL		L5 vs. L6	Q8 vs. Q10	P vs. E	1	5.767	0.02	0.090
CLASS * NLEN * QLEN * QQUAL		L5 vs. L6	Q6 vs. Q8	P vs. E	1	5.747	0.02	0.090

Table 7-11. NREC: Significant tests of within-subject contrasts

Source	CLASS	NLEN	QLEN	QQUAL	df	F	Sig.	Eta Squared
CLASS	C7 vs. C15				1	6.387	0.014	0.099
NLEN		L4 vs. L5			1	16.806	0	0.225
NLEN		L5 vs. L6			1	45.356	0	0.439
QLEN			Q6 vs. Q8		1	4.421	0.04	0.071
QLEN			Q8 vs. Q10		1	25.686	0	0.307
QQUAL				P vs. E	1	117.503	0	0.670
CLASS * NLEN	C7 vs. C15	L4 vs. L5			1	6.861	0.011	0.106
CLASS * NLEN	C7 vs. C15	L5 vs. L6			1	10.105	0.002	0.148
CLASS * QLEN	C7 vs. C15		Q6 vs. Q8		1	7.413	0.009	0.113
CLASS * QQUAL	C7 vs. C15			P vs. E	1	10.22	0.002	0.150
NLEN * QLEN			Q8 vs. Q10		1	10.75	0.002	0.156
NLEN * QQUAL		L4 vs. L5		P vs. E	1	23.003	0	0.284
NLEN * QQUAL		L5 vs. L6		P vs. E	1	46.962	0	0.447
QLEN * QQUAL			Q6 vs. Q8	P vs. E	1	4.492	0.038	0.072
QLEN * QQUAL			Q8 vs. Q10	P vs. E	1	23.802	0	0.291
CLASS * NLEN * QLEN	C7 vs. C15		Q8 vs. Q10		1	5.14	0.027	0.081
CLASS * NLEN * QQUAL	C7 vs. C15	L4 vs. L5		P vs. E	1	5.46	0.023	0.086
CLASS * NLEN * QQUAL	C7 vs. C15	L5 vs. L6		P vs. E	1	9.863	0.003	0.145
CLASS * QLEN * QQUAL	C7 vs. C15		Q6 vs. Q8	P vs. E	1	8.604	0.005	0.129
NLEN * QLEN * QQUAL		L5 vs. L6	Q8 vs. Q10	P vs. E	1	11.965	0.001	0.171

7.3.3 Interpreting the multi-way interactions through QQUAL

The first step toward a meaningful interpretation of the multi-way interactions is to revisit the NPREC results presented in Figure 7-2 (CLASS and QQUAL), Figure 7-3 (NLEN and QQUAL), and Figure 7-4 (QLEN and QQUAL). Note in each chart that, under the *P* condition, the retrieval results are quite respectable and relatively stable with a general trend toward improvement as one moves from the lower to the upper levels of each factor. However, under the *E* condition, the results for each level of these factors are distinct from one another. Within the factor (i.e., within each chart) we see that as the level changes so does the NPREC value for *E*.[70] Across the factors (i.e., between charts) the *E* curve behaves differently under each. Understanding the nature of the within- and

[70] With one exception: the non-significant difference between *C15E* and *CUE*.

between-factor behavior of the *E* curves is the key to interpreting the multi-way interactions, the most important of which, as stated before, involve QQUAL.

Under the CLASS factor (Figure 7-2), comparing the 0.03 difference between the *C7E* (0.58) and *C15E* (0.61) with the 0.05 difference between *C7P* (0.92) and *C15P* (0.97) informs us in two ways. First, it tells us that the oblique divergence of the *P* and *E* slopes is the source of the significant CLASS by QQUAL interaction. Second, with only a 0.03 difference in the *E* slope (between *C7E* and *C15E*, with none between *C15E* and *CUE*), these results also inform us that CLASS contributes very little to fault tolerance. In fact, the *C7E* to *C15E* slope is *positive* which indicates that fault tolerance improves as number of classificatory categories increases. This finding is very surprising because it is exactly opposite to what we had hypothesized. We had originally conjectured that the loss of information caused by application of the *C7* scheme would afford a strong measure of fault tolerance in the spirit of Parsons' (1975) Same-Up-Down scheme (Subsidiary hypothesis 3). We had expected a loss in precision under the *P* condition for *C7* (i.e., a relatively low NPREC value compared to the other CLASS levels) to occur and it did. However, this loss in precision under the *P* condition was supposed to be compensated by its conjectured error-resistance abilities. As the hypothesized fault tolerance does not appear to exist, and given that the *C7P* has an averaged NPREC of 0.92 vs. 0.97 for both *C15P* and *CUP*, there is no reason to recommend the use of the *C7* classification scheme in a MIR system.

Under the NLEN factor (Figure 7-3), we see an *E* curve which slopes downward as one progresses from *L4E* (0.67) to *L5E* (0.64) to *L6E* (0.50), for a drop of 0.17 between extremes. The *P* curve slopes gently upwards. This divergence of slopes gives rise to the significant NLEN by QQUAL interaction. Because the total difference of 0.17 is greater than the 0.03 difference of the CLASS factor, we see that NLEN plays a greater role influencing retrieval performance. This interpretation is confirmed by the greater η^2 values associated with NLEN. We also see that *L6E*, with its distinctly low NPREC value (0.50), should be flagged as providing important information concerning the multi-way interactions.

Under the QLEN factor (Figure 7-4), the *E* curve slopes strongly upward as one progresses from *Q6E* (0.36) to *Q8E* (0.60) to *Q10E* (0.85), for a gain of 0.49 between extremes. The significant QLEN by QQUAL interaction is explained by the strong, oblique convergence of the gently rising *P* curve and the steeply rising *E* curve. That QLEN plays the most influential role in retrieval performance when compared to CLASS and NLEN is evident in the 0.49 (QLEN) vs. 0.17 (NLEN) vs. 0.03 (CLASS) differences between extremes. This is also confirmed by the η^2 values associated with QLEN. We finally note that *Q6E*, with its distinctly inferior 0.36 NPREC value, should also be flagged as providing more information concerning the multi-way interactions.

From these analyses, we see certain key facts emerging, the most important of which involve NLEN and QLEN as they interact with QQUAL. First, under their respective *E* conditions, when NLEN increases NPREC decreases while for QLEN the opposite obtains. Second, the worst performance for each, under the *E* condition, occurs when the n-gram length used in the query (*Q6*) equals the n-gram length used in the index (*L6*). Knowing these two key facts unlocks most of the mystery behind interpreting the multi-way interactions.

7.3.3.1 Understanding the NLEN by QLEN by QQUAL interaction

Notwithstanding that CLASS participates in many statistically significant multi-way interactions, we believe that understanding the NLEN by QLEN by QQUAL interaction is pivotal to understanding the outcome of our retrieval experiments, particularly with regard to making recommendations for MIR system development. We believe so because:

1) the *C7* level of CLASS has been shown to be inferior under the *P* condition;
2) the *C7* level of CLASS has been shown not to provide the fault tolerance under the *E* condition that might compensate for its lower P condition performance;
3) there is no statistically significant difference between *C15* and *CU* which, when combined with our recommendation that *C7* be eliminated from further consideration, makes further evaluation of CLASS a moot point; and,

4) the NLEN by QLEN by QQUAL interaction, with its η^2 value 0.596, is the most influential of all the 3-way interactions.

Why is it that *CUL6P* provides the best performance while *CUL6E* provides the worst? Why is it that *Q6E* has the lowest NPREC value of all (0.36) while *Q10E* has a respectable 0.85? The answer lies in the fact that the fault tolerance that we expected to achieve via the CLASS factor is actually coming about via the complex interaction of:

1) the ranking retrieval methods employed by us under the SMART system; and,
2) the n-gramming process (i.e., NLEN)
3) the length of the query string (i.e., QLEN).

Through another series of imaginary examples, we can illuminate the essential features of this interaction:

Example 5: Imagine that ***abcdef*** represents a *CUQ6P* query (i.e., an unclassified, error-free query of length-6). If this n-gram were submitted against the *CUL6* database, then one would expect to retrieve at the highest ranks, all those songs in which ***abcdef*** occurs. This situation results in a perfect normalized precision score for the query.

Example 6: Next, imagine that the user introduces an error at the second interval, resulting in a deformed *CUQ6E* query of ***aXcdef***. If this n-gram were submitted against the *CUL6* database, one of two situations would occur, neither of which is beneficial. First, ***aXcdef*** might exist as an n-gram in the *CUL6* database resulting in the retrieval at the highest ranks of those songs in which ***aXcdef*** occurs. If ***aXcdef*** and ***abcdef*** happen to co-occur in a song, then the desired songs would also be retrieved at the highest ranks but purely by happenstance. Second, if ***aXcdef*** *does not* exist as an n-gram in the *CUL6* database, then SMART defaults back to ranking the documents by accession number which results in retrieval rankings no more meaningful than random selection. For these reasons, the *CUL6* database performs best with *CUQ6P* queries (Example 5) and worst with *CUQ6E* queries (Example 6).

Example 7: Now imagine a situation at the other end of the spectrum. Imagine that ***abcedefghij*** represents a *CUQ10P* query. Also, imagine that the *CUL4* database is to be searched. Before submission the ***abcedefghij*** query string would be 4-grammed into ***abcd***, ***bcde***, ***cdef***, ***defg***, ***efgh***, ***fghi,*** and ***ghij***. After n-gramming, this collection of 4-grams would be submitted against the databases. One would expect strong retrieval rankings for the desired songs given the number of n-grams submitted. The results would not be necessarily perfect, however. As we have stated before, we did not include within-song location information in the indexes. Therefore, it is possible that all seven of the 4-grams might exist in a given song but the order of the n-grams in that song might not represent a contiguous string as they do in the query string. This would cause such a song to be ranked highly although the n-grams it contains do not represent the same contiguous string as the query. Thus, there is still an opportunity for several false-drops. Notwithstanding this potential for false-drops, the *Q10* data suggest that the lack of within-song information does not have a significantly detrimental effect on retrieval performance (e.g., *Q10P* has an NPREC of 0.98).

Example 8: Finally, imagine the situation above, only this time the user introduces an error at the second position of the query, which results in a deformed ***CUQ10E*** string of ***aXcdefghij.*** Before submission against the *CUL4* database, the ***aXcdefghij*** query string would be 4-grammed into ***aXcd***, ***Xcde***, ***cdef***, ***defg***, ***efgh***, ***fghi***, and ***ghij.*** Note how n-gramming eventually "by-passes" the interval error. This "by-passing" of the error interval results in the retention of five "perfect" 4-grams, and the creation of only two "defective" 4-grams.[71]

With regard to the error-ridden ***aXcd*** and ***Xcde*** 4-grams, one of three scenarios could occur:

---

[71] Similarly**,** 4-gramming ***abcdefghXj*** would result in 4-grams of ***abcd***, ***bcde***, ***cdef***, ***defg***, ***efgh***, ***fghX***, and ***ghXj***. Note that there would remain five perfect, and two defective, n-grams regardless of which end the error interval occurs. Thus, the "by-passing" of the error interval by the n-gramming process is symmetric with regard to the distance from the ends of the query string.

Scenario 1: Both deformed n-grams exist as n-grams in the database

Under this scenario, the ranking of those songs in which the defective n-grams occur would be increased. This would cause the desired songs to have lower, though not necessarily the lowest, retrieval rankings. However, because there are still five perfectly formed n-grams in the collection of seven query n-grams, there remains a strong error-mitigation factor at play even if the deformed n-grams occur in the database.

Scenario 2: Only one of the deformed n-grams exists in the database

Here the level of error-mitigation would increase relative to Scenario 1 because, under the ranking-retrieval model employed by us under SMART, a term that does not occur in the database is assigned a weight of 0. More precisely, because the n-gram does not exist in the database it would have a *tf*idf* value equal to 0. With a weight of 0, it ceases to be influential in the ranking process. This would leave five of the six 4-grams contributing positively to the retrieval rankings, which is better than the five of seven projected under Scenario 1.

Scenario 3: Both of the deformed n-grams do not exist in the database

This scenario would result in a situation where the *Q10E* would be transformed into a *Q8P* query string. This would occur because of the 0-weighting assigned to those n-grams not found in the database, which would leave five 4-grams to contribute positively, and none negatively, to the ranking process. A query with five perfect, and no deformed, 4-grams would be identical to the *Q8P* condition. This reduction to *Q8P* is most certainly not catastrophic because, under our experiments, the *Q8P* condition has a very strong NPREC value of 0.95.

What our scenarios demonstrate is that the n-gramming process has the ability to "by-pass" a single defective interval if it occurs near the ends of the query strings. This

by-passing of a defective interval is contingent upon the interaction of NLEN, QLEN, and the position *x* of the defective interval.[72]

We have determined that the complex interaction of the NLEN, QLEN, and *x*, can be completely described through application of the appropriate conditional expression (ConExp) (i.e., appropriate for the particular combination of NLEN, QLEN, and *x*). By reducing this complex interaction to one of the four conditional expressions below, it is possible to derive general statements of its behavior. Thus, these conditional expressions can be used, in turn, to assist in making design recommendations.

ConExp 1: If QLEN – NLEN = NLEN, then there are *x* defective n-grams until *x* = NLEN. There are conversely QLEN – NLEN – *x* + 1 perfect n-grams.

The *CUL4Q8E* condition falls under this condition because $8_{QLEN} - 4_{NLEN} = 4 = \text{NLEN}$. Thus, if an error occurs at $x = 3$ there are 3 defective n-grams created and $8_{QLEN} - 4_{NLEN} - 3_x + 1 = 2$ perfect n-grams retained. If the error occurs at $x = 4$, then there are 4 defective n-grams created and 1 perfect n-gram retained. If the error occurs at $x = 5$, there are still 4 defective n-grams created (and 1 perfect retained) because $x >$ NLEN and NLEN represents the upper limit to the number of defective n-grams. Thus, we see that error tolerance is negatively affected by increases in *x*. Furthermore, if $x =$ NLEN and QLEN – NLEN = NLEN, the amount of error tolerance is particularly low.

ConExp 2: If QLEN – NLEN >= NLEN + 1 and *x* >= NLEN there are NLEN defective n-grams. Conversely, there are QLEN – 2NLEN +1 perfect n-grams.

The *CUL4Q10E* combination falls under this condition. Until $x = L$, the number of defective n-grams in *CUL4Q10E* equals *x* just as it does in ConExp 1. Again, like ConExp 1, the maximum number of defective n-grams is 4 (i.e., NLEN). Unlike ConExp 1 however, the minimum number of perfect n-grams

---

[72] As measured from either end of the query string, with *x* beginning at 1.

retained is 3, not 1, because $10_{QLEN} - 2*(4_{NLEN}) + 1 = 3$. The minimum number of defective n-grams created is 1 which occurs when $x = 1$. Again, this is the same as in ConExp 1, however, the ratio of perfect to defective n-grams under ConExp 2 is six perfect to one defective, which provides a much greater degree of fault tolerance. Because NLEN represents the upper limit to the number of defective n-grams, if one decided to increase QLEN beyond *Q10*, then the ratio of perfect to defective n-grams would increase directly with any increase in QLEN. Such an increase in QLEN would then make it even more likely that the desired song(s) would be ranked highly despite the presence of an error. Thus, we see that the error tolerance afforded by n-gramming is directly proportional to QLEN and inversely proportional to NLEN. We also see that fault tolerance is greatly enhanced once QLEN increases beyond 2NLEN + 1.

ConExp 3: If QLEN – NLEN < NLEN, then there are *x* defective n-grams until *x* = QLEN – NLEN + 1. Conversely, there are (x – 1) + 1 perfect n-grams until *x* = QLEN – NLEN + 1.

The *CUL6Q8E* combination is governed by this conditional expression. If $x = 1$, then there is 1 defective n-gram created and 2 perfect n-grams retained. If $x = 2$, then there are 2 defective n-grams created and 1 perfect n-gram retained. If $x = 3$, then there are 3 defective n-grams created and no perfect n-grams retained. Because of the limits imposed by QLEN – NLEN + 1, there is no opportunity for the n-gramming process to reach beyond the error interval to start accumulating perfect n-grams. So, unlike the case presented in ConExp 2, the ratio of perfect to deformed n-grams can only be in favor of the perfect n-grams when $x = 1$. This reinforces the point that fault tolerance is best served when QLEN is large and NLEN is small.

ConExp 4: If QLEN = NLEN, then there is one defective n-gram regardless of *x*. Conversely, there is no perfect n-gram created.

The aforementioned *CUL6Q6E* combination is governed by this conditional expression. Because NLEN = QLEN there is no opportunity whatsoever for the n-gramming process to by-pass an erroneous interval. Also,

there is no perfect n-gram created that could compensate for the presence of the error. As stated before, only pure happenstance would place the desired song(s) in a highly ranked position. Thus, if one wants some measure of fault tolerance, this scenario represents, *ad extremis,* the need for QLEN to be greater than NLEN.

To summarize, when a single interval error occurs in a query string (i.e., the *E* level of QQUAL) the complex interaction of:

1) the location *x* of the erroneous interval;
2) the QLEN of the query string in which it is found;
3) the NLEN of the n-gramming process applied to the query string; and,
4) the ranking-method employed by us under SMART in our experiments,

combine to provide varying degrees of fault tolerance. It is this varying degree of fault tolerance that underpins the NLEN by QLEN by QQUAL interaction. In fact, to understand the NLEN by QLEN by QQUAL interaction is to understand the nature of fault tolerance afforded by the n-gramming process. Simply put, the fault tolerance that we have uncovered with respect to our experiments ***is*** the NLEN by QLEN by QQUAL interaction. We conclude, therefore, that the fault tolerance brought about by the interaction of NLEN and QLEN is generally:

1) inversely proportional to the location *x* of the erroneous interval;
2) directly proportional to length of the query string, QLEN, but more precisely its length *relative to* NLEN (i.e., QLEN must not only be greater than NLEN, it should be substantially greater); and,
3) inversely proportional to the length of the n-grams created, NLEN, but more precisely, its length *relative to* QLEN (i.e., NLEN must not only be less than QLEN, it should be substantially less).

Since the location *x* of the defective interval is beyond the control of a system designer, we will not factor it into our design recommendations, which follow below.

7.4 Design recommendations

Because of the significance of the NLEN by QLEN by QQUAL interaction, and the lack of fault tolerance provided by the *C7* level of CLASS, it is impossible to provide a single combination of system factors (i.e., CLASS and NLEN) that would lead to the "best" MIR system under all types of user conditions (i.e., QLEN and QQUAL). However, by revisiting the informal questions posed in Chapter 5.3.4 concerning each of our independent factors, we can provide conditional recommendations with regard to each factor.

7.4.1 Informal questions regarding experimental factors: revisited

Question 1: (CLASS) Does the size of the classificatory set used in the creation of the n-gram representations affect performance?

Yes. Our tests of within-subjects test show that *C7* < (*C15* = *CU*). However, the expected fault tolerance brought about through the application of the classification function does not appear to exist. Thus, there is nothing recommending the use of either the *C7*, or the *C15*, classification schemes. We recommend, therefore, the use of the *CU* approach (i.e., leave the intervals unclassified). This recommendation has the added benefit that the classification process can be eliminated altogether, thus saving preprocessing time.

Question 2: (NLEN) Does the length of the n-gram representations affect performance?

Yes. If one is unconcerned about the possible presence of query errors then we recommend the use of the *L6* n-gram length. With its NPREC value of 0.9927 under the combination of the previously recommended *CU* level of CLASS and the *Perfect* level of QQUAL, *L6* provides the superior performance (averaged over queries of all lengths).

If one wishes to maximize the fault tolerance of a MIR system, then we recommend the *L4* n-gram length. *CUL4E* has a NPREC value of 0.6952, averaged over queries of all lengths, which makes it the superior combination under the *Error* level of the QQUAL factor. See also Question 5 below.

Question 3: (QLEN) How much of the melody will the users have to remember?

If one is unconcerned about the presence of a query error, then the NPREC values of *Q6P* (0.94), *Q8P* (0.95), and *Q10P* (0.98) indicate that shorter queries are not problematic, although longer queries are to be preferred.

If, however, one wishes to maximize the fault tolerance of the system, then it is very important that longer queries be obtained. The distinctly inferior performance of *Q6E* (0.36), when compared to the superior performance of *Q10E* (0.85), reinforces the importance of longer queries to fault tolerance. How one enforces the provision of longer queries, however, is an interface issue beyond the scope of this study. See also Question 5 below.

Question 4: (QLOC) Does the location of the query affect retrieval effectiveness?

No. This finding suggests that information useful for retrieval is distributed throughout each song. We therefore recommend that entire songs be indexed rather than *Incipits*.

Question 5: (QQUAL) Do minor query errors affect performance?

Yes. If one is unconcerned about the presence of a query error then this finding is moot.

If, however, one wishes to provide the maximum amount of fault tolerance it is very important that NLEN be very much shorter than QLEN. Since the only factor truly under the control of a system designer is NLEN, NLEN should be minimized. Thus, we recommend again that MIR indexes should use *L4*.

Note, however, that *CUL4Q6E* has an NPREC of 0.48, which definitely does not represent strong retrieval performance. It is not until *CUL4Q10E*, with its NPREC value of 0.89, that the fault tolerance afforded by the n-gramming process has real implications. Therefore, it is very important to stress that the fault tolerance afforded by n-gramming is highly dependent upon the user

providing a long query. This being the case, with regard to the presence of query errors, we recommend that designers either:

1) in some way enforce long queries; or,

2) develop more effective methods of error detection and correction.

To summarize our recommendations with regard to the factors under the control of a designer (i.e., CLASS and NLEN) we conclude that:

1) if fault tolerance is NOT an issue, then use *CUL6; or,*

2) if fault tolerance is an issue, then use *CUL4* with the proviso that long queries, or some new type of error detection and correction, are required for truly robust performance.

7.5 Salton's space-density *Q* and NPREC

In Chapter 6.3.4 we undertook a term discrimination analysis in an effort to ascertain, in part, what the space density *Q* values could tell us about the potential retrieval performances of our music databases. Since Salton (1975) explicitly claimed that a database's precision results would be inversely proportional to its *Q* value, we decided to drop *C3* from further consideration, in part, because of its high average *Q* (0.44). We also noted that *C7L4* appeared to have a questionable *Q* value of 0.23, while the remaining databases appeared to have *Q* values suggestive of strong precision performances. Implicit in this decision were our assumptions that a) Salton's claims concerning precision and document space-density are correct; and, b) Salton's theory can be applied to our non-textual, folksong databases.

In an effort to determine whether our assumptions concerning the *Q* analysis were well-grounded, we decided to regress *Q* on NPREC. Because Salton makes no claims about precision in the presence of errors, we limited our regression analysis to the *Perfect* condition NPREC data. We have determined that the following linear model (significant at $p < 0.002$, with 1 and 7 degrees of freedom) can be used to describe the relationship between *Q* and NPREC:

$$NPREC_{(Perfect)} = 1.01 - 0.47 * Q$$

The above model has a Multiple R value of 0.88 and an R^2 value of 0.77. R^2, like η^2, ranges in value from 0 to 1 and measures the "proportion of variation in the dependent variable explained by the regression model" (SPSS 1977). Since the model's R^2 is a strong 0.77, we have concluded that our assumptions concerning the use of *Q* appear to be well-grounded.

Furthermore, the model provides us with a general informetric model with which we can describe our present results, and the results of any future experiments. Specifically, those n-gram databases with lower *Q* values have, and will have, the higher NPREC results. Thus, in the future, should we decide to explore new n-gram lengths or explore different classification schemes we can use the *Q* values associated with the new database configurations to determine whether retrieval evaluations would be warranted.

7.6 Summary and conclusions

In this chapter, we have reported upon our Phase II IR simulations and evaluations. Through the use of a complex, mixed-, five-way factorial experimental design, we set out to ascertain whether our independent factors CLASS, NLEN, QLEN, QQUAL, and QLOC, play significant roles in the retrieval performance of our n-grammed music databases as measured by NPREC. Analyzed with a repeated-measures MANOVA, which included the application of *a priori* within-subject contrast codes, we have determined that:

1) CLASS, NLEN, QLEN, and QQUAL all influence retrieval performance;
2) the strength of the NLEN by QLEN by QQUAL interaction makes the recommendation of a single "best" database configuration impossible;
3) the *C7* and *C15* classification schemes failed to provide any useful protection against the presence of single-interval query errors;
4) the n-gramming process provides a useful level of fault tolerance when NLEN is small (i.e., *L4*) and *QLEN* (i.e., *Q10*) is large but when QLEN approaches NLEN, the level of fault tolerance greatly diminishes;

5) the augmented *tf * idf* ranking method employed affords strong retrieval performance, notwithstanding the lack of within-song location information; and,

6) we recommend *CUL4* for those concerned about query errors, while we recommend *CUL6* for those not so concerned.

We have also determined that the data provided through the Phase I informetric analyses proved themselves both useful and reliable in three important ways. First, given the strong performances of our databases under the *P* condition, it appears that our choice of ranking method, a decision based upon our interpretation of the informetric properties of our databases, was sound. While we cannot claim to have made the optimal selection of ranking method, we can claim that our choice did not demonstrate any obviously pathological characteristics. Second, the informetric data consistently, and correctly, suggested that *C7* < (*C15* = *CU*). Third, the linear regression analysis of the *Q* space-density data provided a useful model for understanding and predicting the retrieval performances of our present and future database configurations (under the *P* condition).

Notwithstanding our interest in the specific behaviors of our independent factors, we conducted the Phase II evaluation to determine the merits of our principal and subsidiary hypotheses. We now conclude our IR evaluation by revisiting these hypotheses.

7.6.1 Principal hypothesis: revisited

It was hypothesized that, for purposes of information retrieval, there is enough information contained within the interval-only representation of monophonic melodies that the n-gramming of interval-only melodic strings into "musical words" and their subsequent indexing will allow users the same access to melodic information that indexes of "real words" give to textual information. [73]

We accept our principal hypothesis. Nothing in our evaluation indicates that our recommended *CUL4* and *CUL6* databases would not perform well in a text-based

---

[73] There are, of course, qualifications that place limitations upon our claims which follow. See Chapter 8 for our enumeration of these qualifications and limitations.

retrieval environment. To the contrary, averaged over queries of all lengths, the NPREC values of *CUL4P* (0.9445) and *CUL6P* (0.9727) suggest that users should experience retrieval performances as strong as, if not stronger than, they experience with "real word" text databases. Also, inherent in the n-gramming process is a modest degree of fault tolerance not present in most textual IR systems, which further supports our acceptance of this hypothesis.

7.6.2 Subsidiary hypothesis 1: revisited

It was hypothesized that there is some type of equivalency between interval-only melodic n-grams (i.e., "musical words") and "real words," intervals and letters.

In Chapter 6, we decided that the hypothesized equivalency between intervals and letters was untenable. We also decided that the hypothesized equivalency between melodic n-grams and words was acceptable only on a metaphoric and pragmatic level. After the IR evaluations, we continue to hold that, on a metaphoric and pragmatic level, viewing melodic n-grams as artificially created "words" has great conceptual merit. The merit of conceptualizing n-grams as words is manifest in the successful application of traditional text-based IR theory (e.g., the strong predictive characteristics of our Phase I informetric analyses; the modeling of system performance based on Salton's *Q*). It is also manifest in the successful application of traditional text-based IR methods (e.g., the "off-the-shelf" use of SMART; the use of augmented *tf * idf* as a ranking method).

We wish to stress that our conceptualization of melodic n-grams as "words" has led us to demonstrate formally that the MIR problem can be reconceived as a traditional IR retrieval problem (i.e., Paradigm 1). Since it can be reconceived as a traditional text-based IR problem, we can now turn to the vast body of research into text-based IR systems for potential solutions to MIR system development. This is not to say that, at some juncture, the pragmatic and metaphoric nature of our "musical words" approach will not break down. However, for the immediate future, we believe that significant improvements can be made through the judicious use of proven text-based theories and methods. Simply put, we have not yet seen a reason to "reinvent" IR theories and methods to solve the MIR problem.

7.6.3 Subsidiary hypothesis 2: revisited

It was hypothesized that the use of some type of ranked retrieval method would overcome any loss of retrieval effectiveness associated with the absence of within-song location information, making it unnecessary to include such information within the indexes.

We accept our second subsidiary hypothesis. Again, the strong NPREC results associated with our recommended database configurations indicate that the lack of within-song location information has little detrimental effect on retrieval performance. This finding, more than any other, indicates that the MIR problem is not necessarily a string-matching problem. This being said, the computationally more expensive approximate-string matching algorithms favored by other research teams, while possibly providing more optimal results, seem unnecessarily complicated and expensive for the amount of enhancement they might provide. In fact, determining how much better the other approaches are when compared to our simple MusiFind approach is impossible given the paucity of formal, standardized, and rigorous evaluations of the other approaches. For more commentary on this issue, see our concluding remarks below.

7.6.4 Subsidiary hypothesis 3: revisited

It was hypothesized that application of the classification function C would offer a level of "forgiveness" (i.e., resilience to query errors) inversely proportional to the number of classes used to classify the intervals.

We reject our third subsidiary hypothesis. Our findings are exactly opposite to our hypothesis: the *C7* classification scheme (averaged over queries of all lengths) actually has a negative effect on fault tolerance. *C15* was not significantly different from *CU*, which again indicates that the classification functions used in this study do not afford the fault tolerance that they were intended to afford.

7.6.5 Concluding remarks

As the title of this thesis states, we have evaluated a simple approach to music information retrieval. We believe that we have shown that our simple approach works. Given the simplicity and success of the MusiFind approach we are now confident that the incorporation of melodic n-grams into standard bibliographic records for use in

traditional full-text bibliographic information retrieval (FBIR) systems should prove successful (Paradigm 1, adoption of the FBIR model). Incorporating melodic n-grams into web pages for indexing under the various World Wide Web search-engines is another option that shows promise. To achieve these incorporations only minor interface issues would need to be addressed. This task should not prove too onerous since both McNab et al. (1996, 1997) and Prechelt and Typke (1998) have developed robust MIR interfaces.

While we most certainly believe that different approaches to MIR system development offer the promise of superior performance, we also believe that the decision to accept those approaches should be based upon the comparison of those approaches with our simple approach (Paradigm 2, the principle of parsimony). We hope, that this thesis, when considered *in toto*, convinces other researchers of the need to standardize MIR evaluations under the Cranfield model. In keeping with this sentiment, we also hope that other researchers will come to see our results as a baseline against which other, more sophisticated approaches, can be formally evaluated. We further hope that other researchers will see merit in the proposition that the decision to accept the more complex solutions should be grounded in the idea that the benefits of those solutions must truly outweigh their costs (i.e., apply the principle of parsimony).

8 Conclusion

The primary objective of this study was to determine whether a simple approach to MIR, one grounded in the notion that MIR could be reconceived as a text-based IR problem, showed merit. Specifically, our research attempted to conclude whether the n-gramming of interval-only monophonic melodic strings could create length-n sub-strings containing enough information that their use as artificial "musical words" would allow them to substitute for "real words" in the context of a text-based IR system (i.e., the MusiFind approach).

Two interrelated operating paradigms governed the entire research endeavour. Paradigm 1 was the adoption of a full-text, bibliographic information retrieval (FBIR) model to the retrieval of music information. Paradigm 2 was the principle of parsimony. This principle led us to hold that a simpler approach to MIR development is to be preferred over a more complex approach unless a more complex approach can be proven to provide significantly better results. Taken together these two paradigms further led us to formally evaluate, under the Cranfield model, our simple MIR approach using an "off-the-shelf" text-based FBIR system (SMART).

Our study examined five independent factors: CLASS, NLEN, QLEN, QLOC, and QQUAL. The CLASS factor was examined to see if the collapsing of information into smaller classes would afford a level of fault tolerance inversely proportional to the number of classes created. NLEN, the length of the n-gram sub-string, was examined to determine if the choice of n-gram length influenced performance. QLEN, the length of the query string, was examined to see what effect query length had on performance. QLOC, the location of the query string, was examined to ascertain whether users could construct queries that represent melodic information found anywhere within a song. QQUAL, the quality of the query (i.e., the presence or absence of an error), was examined to ascertain whether the CLASS factor afforded the hypothesized fault tolerance.

To ascertain whether the MusiFind approach would work, we undertook a two-phased evaluation, loosely modeled on the thesis research of Wolfram (1992a and

1992b). Phase 1 was a wide-ranging set of informetric analyses intended to determine the informetric properties of the twelve n-grammed music databases created under various combinations of CLASS and NLEN. Phase II, based upon the Cranfield model of evaluation, was the formal IR simulation and evaluation experiment. Phase II evaluated the influence of CLASS, NLEN, QLEN, QLOC and QQUAL on retrieval performance, as measured by NPREC and NREC, with NPREC being the primary determinant.

Key findings of the Phase I informetric analyses include:

1) Databases created under certain combinations of CLASS and NLEN have informetric properties that suggested further investigation would be unwarranted. Specifically, the extreme reduction in information caused by the *C3* CLASS makes the databases created under this condition unsuitable for implementation.

2) The *C7*, *C15*, and *CU* databases have informetric properties suggestive of adequate (*C7*) or superior (*C15* and *CU*) information retrieval performance. More precisely, these databases did not exhibit any pathological informetric characteristics that would preclude their use in our IR simulation and evaluation.

3) The *C7*, *C15*, and *CU* databases are naturally "stopword-free."

4) The *C7*, *C15*, and *CU* databases exhibit informetric characteristics more closely associated with document surrogate databases than traditional full-text databases.

5) The hypothesized equivalency between "musical words" and "real words" is best conceived as being only metaphorically and pragmatically true. More precisely, there are some significant differences between the two but conceptually the equivalency has shown itself a useful construct.

The principal findings of the Phase II simulation and evaluation include:

1) The hypothesized fault tolerance associated with the CLASS factor does not appear to exist. We therefore recommend that the *CU* condition be used.

2) Fault tolerance is afforded by the interaction of NLEN and QLEN. Specifically, if QLEN is substantially greater than NLEN, then the n-gramming process has the ability to "by-pass" the error, thus creating a sufficient number of "perfect" n-grams that can compensate for the presence of the "defective" n-grams created by the error.

3) There is a complex interaction between NLEN, QLEN, and QQUAL, which makes the recommendation of a single "best" combination of CLASS and NLEN impossible. For situations where fault tolerance is desired, we recommend *CUL4*. For situations where fault tolerance is deemed unnecessary, we recommend *CUL6*.

4) Retrieval performance can be modeled as being inversely proportional to the document space-density *Q* values of the n-grammed databases. This finding provides a useful tool for future research in that one can now predict the relative retrieval performances of novel database configurations prior to experimentation. Thus, by examining only those novel configurations with the superior *Q* values (i.e., lower Q values), one can save valuable research time.

8.1 Limitations of the study

As we have loosely modeled our endeavours on the work of Wolfram (1992a, 1992b), our study can be subjected to some of the same criticisms as his. The principal criticism concerning IR simulations is that the lack of human subjects renders them "too artificial" to be of any use in the "real-world." Wolfram (1992b) successfully dealt with this criticism when he stated:

> …it can be argued that the results may not be generalizable because systems were examined under simulated circumstances and, admittedly, under ideal conditions. Simulation models cannot take into account all the possible variables and intricacies that may come into play in a real-world situation; therefore assumptions and simplifications must be made. With appropriate models such studies can still provide significant insights into systems behavior.

The size of database examined in this study is another valid criticism that can be raised. In this age of giga-byte and tera-byte databases, the relatively small size of our

music database makes extrapolation of our results problematic. Notwithstanding the validity of this criticism, we believe that we have shown that the MusiFind approach could be used as an efficient first-stage filter for a hybrid MIR system that combined our methods with one of the sophisticated string-matching algorithms.

The folksong genre of music examined in our study also imposes some significant limitations that must be addressed. Folksongs are both short and simple, almost by definition. Because folksongs tend to be short and contain a limited variety of intervals, the informetric properties uncovered, and the retrieval results obtained, are most likely not generalizable outside the domain of simpler vocal music. However, since there exists a vast repertoire of simpler vocal music (e.g., folksongs, popular songs, choral music, hymns, and anthems, etc.) the findings of this study are far from trivial.

The lack of rhythm and polyphony information in our n-grammed databases is another shortcoming. Although we explicitly set out to evaluate a simple approach to music information, it must be acknowledged that both rhythm and polyphony are so fundamental to the nature of music that any system which ignores these facets of music information is far from ideal. In short, simple might be good, but it is by no means best.

8.2 Future research

Out of the criticisms above comes a rich MIR research agenda.

Human subjects must be brought into the evaluation of the MusiFind approach. Specifically, data must be gathered and analyzed with regard to the queries posed by human subjects. Query length and the nature of query errors must be rigorously examined. Given the importance of the NLEN by QLEN by QQUAL interaction, any future advancement made under the MusiFind approach will most likely be predicated upon the knowledge gained from careful analyses of “real-world” human-system interaction.

The evaluation of the MusiFind approach within the context of a large-scale collection is another project that must be undertaken. Should such an evaluation prove successful, then work should be undertaken to test the proposition that Internet search engines might be able to index and retrieve webpages containing melodic n-grams. Under this scenario, significantly larger collections of music could be indexed and

retrieved than otherwise might be the case because the work involved in encoding the music information could be distributed among all those interested in providing access to their collections. Should a large-scale evaluation prove unsuccessful, then work should be undertaken to test the proposition that the MusiFind approach would make an efficient first-stage filter for the more sophisticated string-matching techniques.

The MusiFind approach must be evaluated with databases containing a wider variety of music. It is more than likely that a "one-approach-fits-all" situation does not apply to MIR. By examining the MusiFind approach with a wide variety of music genres, it should be possible to ascertain for which genres the MusiFind approach is applicable and for which genres other methods are to be recommended.

Research concerning the extension of the MusiFind approach to incorporate some type(s) of rhythm and polyphony information must also be undertaken. Research regarding a rhythm component would also be predicated on the behavior of users. Therefore, again, there is a further need to conduct appropriately designed human-system interaction experiments.

Given the complex interaction of pitch, interval, and rhythm that constitutes polyphony, the incorporation of a polyphony component is particularly problematic. We believe that the "polyphony problem" will prove to be the most intractable of all the problems associated with MIR development. Since intractable problems are the stuff of research, we foresee a long, busy future ahead for MIR researchers and developers.

.

Appendix A: Rejected Model Fitting Data

Table A-1. Rejected View A model fitting data

Zipf							
Scheme	***a***	**b**		X^2	**d.f.**	**Critical X^2**	**Decision**
C7L4	0.09	0.92		1126.90	337	381	REJECT
C7L5	0.22	1.14		4107.91	616	675	REJECT
C7L6	0.38	1.42		9495.99	536	591	REJECT
C15L4	0.29	1.31		1602.24	450	500	REJECT
C15L5	0.43	1.54		3626.29	465	516	REJECT
C15L6	0.56	1.83		4985.69	312	354	REJECT
CUL4	0.38	1.48		851.05	367	413	REJECT
CUL5	0.48	1.67		1762.88	372	418	REJECT
CUL6	0.59	1.92		2998.97	264	303	REJECT
Generalized Waring (Zero-Truncated)							
Scheme	**a**	**β**	**ν**	X^2	**d.f.**	**Critical X^2**	**Decision**
C7L4	0.31	0.43	10.25	1674.22	190	223	REJECT
C7L5	1.01	0.50	23.73	685.86	314	356	REJECT
C7L6	1.90	0.34	13.75	216.29	172	204	REJECT
C15L4	0.79	0.17	11.78	588.32	275	315	REJECT
C15L5	1.31	-0.03	11.26	404.73	217	252	REJECT
C15L6	2.08	-0.20	9.83	450.25	119	145	REJECT
CUL4	0.83	-0.19	16.05	420.52	264	303	REJECT
CUL5	1.34	-0.26	15.52	453.29	210	245	REJECT
CUL6	2.22	-0.32	12.56	634.44	111	137	REJECT
Mandelbrot-Zipf							
Scheme	***a***	**b**		X^2	**d.f.**	**Critical X^2**	**Decision**
C7L4	0.31	0.43	10.25	1674.22	190	223	REJECT
C7L5	0.76	1.41	2.80	1625.68	456	507	REJECT
C7L6	10.98	2.37	4.06	452.79	207	242	REJECT
C15L4	0.80	1.54	1.41	724.19	338	382	REJECT
C15L5	1.72	1.95	1.29	623.47	266	305	REJECT
C15L6	2.58	2.40	0.99	355.80	158	188	REJECT
CUL4	0.89	1.70	0.81	555.95	272	311	REJECT
CUL5	1.00	1.90	0.55	563.95	266	305	REJECT
CUL6	1.72	2.33	0.64	404.48	159	189	REJECT

Table A-2. Rejected View B model fitting data

Zipf							
Scheme	***a***	***b***		X^2	**d.f.**	**Critical X^2**	**Decision**
C7L4	0.10	0.92		1041.12	308	350	REJECT
C7L5	0.23	1.14		3418.74	503	556	REJECT
C7L6	0.40	1.45		7208.49	356	401	REJECT
C15L4	0.31	1.33		1147.69	384	431	REJECT
C15L5	0.45	1.59		2076.52	341	385	REJECT
C15L6	0.60	1.95		1951.92	200	234	REJECT
CUL4	0.41	1.52		558.19	300	341	REJECT
CUL5	0.51	1.75		781.25	269	308	REJECT
CUL6	0.63	2.06		810.09	172	204	REJECT
Generalized Waring (Zero-Truncated)							
Scheme	**a**	**β**	**ν**	X^2	**d.f.**	**Critical X^2**	**Decision**
CL74	0.72	1.30	22.05	711.66	220	256	REJECT
C7L5	1.21	0.85	13.87	748.07	236	273	REJECT
C7L6	2.05	0.22	15.42	194.02	148	177	REJECT
C15L4	0.84	-0.02	21.38	492.30	249	287	REJECT
C15L5	1.32	-0.14	11.84	312.46	191	224	REJECT
C15L6	1.88	-0.31	7.50	185.93	111	137	REJECT
CUL4	0.88	-0.26	18.37	425.51	226	262	REJECT
CUL5	1.43	-0.38	23.20	270.39	183	216	REJECT
CUL6	2.06	-0.46	13.15	198.27	107	132	REJECT
Mandelbrot-Zipf							
Scheme	***a***	***b***	***c***	X^2	**d.f.**	**Critical X^2**	**Decision**
C7L4	0.29	1.10	6.39	673.88	280	320	REJECT
C7L5	2.70	1.70	5.77	857.02	321	364	REJECT
C7L6	8.37	2.36	3.27	447.42	177	209	REJECT
C15L4	0.86	1.58	1.27	699.07	286	326	REJECT
C15L5	1.05	1.86	0.70	650.99	254	292	REJECT
C15L6	1.51	2.31	0.53	435.33	140	169	REJECT
CUL4	0.94	1.76	0.71	774.93	226	262	REJECT
CUL5	0.84	1.93	0.31	669.97	213	248	REJECT
CUL6	0.90	2.20	0.19	488.35	149	178	REJECT

Appendix B: Within-Subject Tests

Table B-1. NPREC: Multivariate tests of within-subject effects

Effect	F	Hypothesis df	Error df	Sig.	Eta Squared
CLASS	12.774	2	57	0	0.309
CLASS * QLOC	0.635	2	57	0.533	0.022
NLEN	16.571	2	57	0	0.368
NLEN * QLOC	0.018	2	57	0.982	0.001
QLEN	89.165	2	57	0	0.758
QLEN * QLOC	0.619	2	57	0.542	0.021
QQUAL	442.557	1	58	0	0.884
QQUAL * QLOC	0.260	1	58	0.612	0.004
CLASS * NLEN	5.368	4	55	0.001	0.281
CLASS * NLEN * QLOC	1.072	4	55	0.379	0.072
CLASS * QLEN	0.569	4	55	0.686	0.040
CLASS * QLEN * QLOC	1.196	4	55	0.323	0.080
CLASS * QQUAL	10.658	2	57	0	0.272
CLASS * QQUAL * QLOC	0.526	2	57	0.594	0.018
NLEN * QLEN	3.445	4	55	0.014	0.200
NLEN * QLEN * QLOC	0.720	4	55	0.582	0.05
NLEN * QQUAL	79.143	2	57	0	0.735
NLEN * QQUAL * QLOC	0.084	2	57	0.919	0.003
QLEN * QQUAL	53.788	2	57	0	0.654
QLEN * QQUAL * QLOC	0.294	2	57	0.746	0.010
CLASS * NLEN * QLEN	1.669	8	51	0.129	0.208
CLASS * NLEN * QLEN * QLOC	1.787	8	51	0.101	0.219
CLASS * NLEN * QQUAL	13.606	4	55	0	0.497
CLASS * NLEN * QQUAL * QLOC	0.250	4	55	0.908	0.018
CLASS * QLEN * QQUAL	14.396	4	55	0	0.511
CLASS * QLEN * QQUAL * QLOC	1.543	4	55	0.203	0.101
NLEN * QLEN * QQUAL	20.294	4	55	0	0.596
NLEN * QLEN * QQUAL * QLOC	0.407	4	55	0.803	0.029
CLASS * NLEN * QLEN * QQUAL	2.840	8	51	0.011	0.308
CLASS * NLEN * QLEN * QQUAL * QLOC	1.411	8	51	0.215	0.181

Table B-2. NREC: Multivariate tests of within-subject effects

Effect	F	Hypothesis df	Error df	Sig.	Eta Squared
CLASS	3.772	2	57	0.029	0.117
CLASS * QLOC	1.015	2	57	0.369	0.034
NLEN	36.102	2	57	0	0.559
NLEN * QLOC	0.589	2	57	0.558	0.020
QLEN	22.634	2	57	0	0.443
QLEN * QLOC	0.213	2	57	0.809	0.007
QQUAL	117.503	1	58	0	0.670
QQUAL * QLOC	0.817	1	58	0.37	0.014
CLASS * NLEN	6.477	4	55	0	0.320
CLASS * NLEN * QLOC	0.711	4	55	0.588	0.049
CLASS * QLEN	2.812	4	55	0.034	0.170
CLASS * QLEN * QLOC	0.776	4	55	0.546	0.053
CLASS * QQUAL	5.539	2	57	0.006	0.163
CLASS * QQUAL * QLOC	1.276	2	57	0.287	0.043
NLEN * QLEN	6.726	4	55	0	0.328
NLEN * QLEN * QLOC	0.120	4	55	0.975	0.009
NLEN * QQUAL	41.384	2	57	0	0.592
NLEN * QQUAL * QLOC	0.615	2	57	0.544	0.021
QLEN * QQUAL	21.252	2	57	0	0.427
QLEN * QQUAL * QLOC	0.255	2	57	0.776	0.009
CLASS * NLEN * QLEN	2.063	8	51	0.057	0.245
CLASS * NLEN * QLEN * QLOC	1.041	8	51	0.419	0.140
CLASS * NLEN * QQUAL	5.999	4	55	0	0.304
CLASS * NLEN * QQUAL * QLOC	0.601	4	55	0.664	0.042
CLASS * QLEN * QQUAL	3.372	4	55	0.015	0.197
CLASS * QLEN * QQUAL * QLOC	0.690	4	55	0.602	0.048
NLEN * QLEN * QQUAL	7.794	4	55	0	0.362
NLEN * QLEN * QQUAL * QLOC	0.115	4	55	0.977	0.008
CLASS * NLEN * QLEN * QQUAL	1.818	8	51	0.095	0.222
CLASS * NLEN * QLEN * QQUAL * QLOC	0.814	8	51	0.594	0.113

Bibliography

MusiFind Publications

Downie, J. Stephen. 1993a. Creating a multi-purpose full text music database. Presented at "*SLIS Colloquium*". *School of Library and Information Science, University of Western Ontario, 3 February 1993, London, Ontario*.

Downie, J. Stephen. 1993b. *Creating the ideal full-text music database: A research report*. Technical report. Graduate School of Library and Information Science, University of Western Ontario. London, Ont.: University of Western Ontario.

Downie, J. Stephen. 1993c. The MusiFind (Music Information Retrieval) Project: A summary of findings. Presented at *The 1993 Regional Graduate Music Student Colloquium, 30 October 1993, University of Toronto, Toronto, Ontario*.

Downie, J. Stephen. 1993d. *Creating the ideal full-text music database: User assessment survey*. Technical report. Graduate School of Library and Information Science, University of Western Ontario. London, Ont.: University of Western Ontario.

Downie, J. Stephen. 1994. The MusiFind Musical Information Retrieval Project, Phase II: User assessment survey. In *The information industry in transition: Proceedings of the 22nd annual conference of the Canadian Association for Information Science, 25-27 May 1994 Montreal, Quebec*, 149-166. Toronto: Canadian Association for Information Science.

Downie, J. Stephen. 1995. The MusiFind Music Information Retrieval Project, Phase III: Evaluation of indexing options. In *Connectedness: Information, systems, people, organizations: Proceedings of the 23rd annual conference of the Canadian Association for Information Science, 7-10 June 1995, Edmonton, Alberta,* 135-146. Toronto: Canadian Association for Information Science.

Downie, J. Stephen. 1996a. Representing melodies as text: Implications for information retrieval. Poster presented at *ALISE '96, 11-14 January 1996, San Antonio, Texas*.

Downie, J. Stephen. 1996b. Toward the creation of a full-text music information retrieval system: A presentation of findings and future directions. Presented at "*GSLIS Colloquium"*. *School of Library and Information Science, University of Western Ontario, 22 May 1996, London, Ontario*.

Downie, J. Stephen. 1997. Informetrics and music information retrieval. In *Communication and information in context: Society, technology, and the professions: Proceedings of the 25th annual conference of the Canadian Association for Information Science, 8-10 June 1997, St. John's, Newfoundland*, 295-308. Toronto: Canadian Association for Information Science.

Downie, J. Stephen. 1998. Informetrics and music information retrieval: An informetric examination of a folksong database. In *Information science at the dawn of the new millennium: Proceedings of the 26th annual conference of the Canadian Association for Information Science, 3-5 June 1998, Ottawa, Ontario*, 375-392. Toronto: Canadian Association for Information Science.

Downie, J. Stephen. 1999a. Representing melodies as collections of "musical words": It works! Poster presented at *ALISE '99, 26-29 January 1999, Philadelphia, PA.*

Downie, J. Stephen. 1999b (accepted). Music retrieval as text retrieval: Simple yet effective. In *Proceedings of the Association for Computing Machinery, SIGIR '99 conference, University of California at Berkeley, 15-19 August 1999, Berkeley, California*. New York: Association for Computing Machinery.

.Tague-Sutcliffe, Jean, J. Stephen Downie, and Shane Dunne. 1993. Name that tune: An introduction to musical information retrieval. In *Information as a global commodity: Communication, processing and use: Proceedings of the 21st annual conference of the Canadian Association for Information Science, 12-14 July 1993, Antigonish, Nova Scotia*, 204-216. Toronto: Canadian Association for Information Science.

General Publications

Barlow, Harold, and Sam Morgenstern. 1949. *A dictionary of musical themes*. London: Ernest Benn.

Brook, Barry S. 1980. Thematic catalogue. In *The new Grove dictionary of music and musicians*, ed. Stanley Sadie. London: Macmillan Publishers.

Brook, Barry S., and Murray J. Gould. 1964. Notating music with ordinary typewriter characters (A Plaine and Easie code for Musicke). *Fontes Artis Musicae* 11: 142.

Burgin, Robert. 1991. The effect of indexing exhaustivity on retrieval performance. *Information Processing and Management* 27 (6): 623-628.

Burrell, Quentin L., and Michael R. Fenton. 1993. Yes, the GIGP really does work—and is workable! *Journal of the American Society for Information Science* 44 (2): 61-69.

Camilleri, Lelio. 1992. The *Lieder* of Karl Collan. *Computing in Musicology* 8: 67-68.

Cleverdon, Cyril, Jack Mills, and Michael Keen. 1966. *Factors determining the performance of indexing systems*. Cranfield, UK: Aslib Cranfield Research Project, College of Aeronautics.

Crawford, Tim, Costas S. Iliopoulos, and Rajeev Raman. 1998. String-matching techniques for musical similarity and melodic recognition. *Computing in Musicology* 11: 73-100.

Crouch, Carolyn J. 1988. An analysis of approximate versus exact discrimination values. *Information Processing and Management* 24 (1): 5-16.

Dewey, G. 1950. *Relativ(sic) frequency of english speech sounds*. Cambridge, MA: Harvard University Press.

Dowling, W. Jay. 1978. Scale and contour: Two components of a theory of memory for melodies. *Psychological Review* 85 (4): 341-354.

Dubin, David. 1997. *TDV.pl*. Unpublished computer programme.

Duggan, Mary K. 1989. CD-ROM, music libraries, present and future. *Fontes Artis Musicae* 36 (2): 84-89.

Duggan, Mary K. 1992. Electronic information and applications in musicology and music theory. *Library Trends* 40 (4): 756-780.

Dunne, Shane. 1993. Some thoughts on music database indexing and query mechanisms. Private correspondence to author, 4 February 1993. 16 pp.

Edson, Jean Slater. 1970. *Organ-preludes: An index to compositions on hymn tunes, chorales, plainsong melodies, gregorian tunes and carols*. Metuchen, NJ: Scarecrow Press.

Egghe, Leo, and Ronald Rousseau. 1990. *Introduction to informetrics: Quantitative methods in library and information science*. Amsterdam: Elsevier Science Publishers.

Fenske, David. 1988. Online Computer Library Center. In *Directory of computer assisted research in musicology 1988*, ed. Walter B. Hewlett and Eleanor Selfridge-Field, 30-31. Menlo Park: Center for Computer Assisted Research in the Humanities.

Fox, Christopher. 1992. Lexical analysis and stoplists. In *Information retrieval: Data structures & algorithms*, ed. William B. Frakes and Ricardo Baeza-Yates, 102-30. Englewood Cliffs: Prentice Hall.

Ghias, Asif, Jonathan Logan, David Chamberlin, and Brian C. Smith. 1995. Query by humming: Musical information retrieval in an audio database. In *Proceedings of the ACM international multimedia conference & exhibition 1995, San Francisco, California*, 231-236. New York: Association for Computing Machinery.

Gringnetti, M.C. 1964. A note on the entropy of words in printed English. *Information and Control* 7 (1): 304-306.

Harman, Donna. 1992. Ranking algorithms. In *Information retrieval: Data structures & algorithms*, ed. William B. Frakes and Ricardo Baeza-Yates, 363-392. Englewood Cliffs: Prentice Hall.

Harman, Donna. 1995. Overview of the Second Text Retrieval Conference (TREC-2). *Information Processing and Management* 31 (3): 271-289.

Harter, Stephen P., and Carol A. Hert. 1997. Evaluation of information retrieval systems: Approaches, issues, and methods. *Annual Review of Information Science and Technology* 32: 3-91.

Haus, Goffredo. 1994. The LIM intelligent music workstation. *Computing in Musicology* 9: 70-73.

Hawley, M. 1990. The personal orchestra, or audio data compression by 10000:1. *Computing Systems* 3 (2): 289-329.

Heaps, H. S. 1978. *Information retrieval: Computational and theoretical aspects*. New York: Academic Press.

Hewlett, Walter B. 1996. A derivative database format for high-speed searches. *Computing in Musicology* 10: 131-142.

Hewlett, Walter B., and Eleanor Selfridge-Field, eds. 1988. *Directory of computer assisted research in musicology.* Menlo Park, CA: Center for Computer Assisted Research in the Humanities.

Hewlett, Walter B., and Eleanor Selfridge-Field, eds. 1990. *Computing in musicology: A directory of research.* Vol. 6. Menlo Park: Center for Computer Assisted Research in the Humanities.

Hewlett, Walter B., and Eleanor Selfridge-Field, eds. 1991. *Computing in musicology: A directory of research* Vol. 7. Menlo Park: Center for Computer Assisted Research in the Humanities.

Hewlett, Walter B., and Eleanor Selfridge-Field, eds. 1992. *Computing in musicology: An international directory of applications.* Vol. 8. Menlo Park: Center for Computer Assisted Research in the Humanities.

Hewlett, Walter B., and Eleanor Selfridge-Field, eds. 1998. *Computing in musicology.* Vol. 11, *Melodic similarity: Concepts, procedures, and applications.* Menlo Park: Center for Computer Assisted Research in the Humanities.

Howard, John B. 1998. Strategies for sorting melodic incipits. *Computing in Musicology* 11: 119-128.

Howard, John, and Joachim Schlichte. 1988. Repertoire international des sources musicales (*RISM*). In *Directory of computer assisted research in musicology 1988*, ed. Walter B. Hewlett and Eleanor Selfridge-Field, 11-24. Menlo Park: Center for Computer Assisted Research in the Humanities.

Huron, David. 1991. *Humdrum*: Music tools for UNIX systems. *Computing in Musicology* 7: 66-67.

Huron, David. 1999. Private email correspondence.

Judd, Charles M., and Gary H. McClelland. 1989. *Data analysis: A model-comparison approach*. Toronto: Harcourt Brace Jovanovich.

Kassler, Michael. 1966. Toward musical information retrieval. *Perspectives of New Music* 4 (Spring-Summer): 59-67.

Kassler, Michael. 1970. MIR: A simple programming language for musical information retrieval. In *The Computer and Music*, ed. Harry B. Lincoln, 299-327. Ithaca, NY: Cornell University Press.

Keen, E. Michael. 1992. Presenting results of experimental retrieval comparisons. *Information Processing and Management* 28 (4): 491-502.

Keller, Kate van Winkle, and Carolyn Rabson. 1980. *National tune index, 18th century secular music.* New York: University Music Edition.

King, A. Hyatt. 1954. The past, present, and future of the thematic catalogue. *Monthly Musical Record* 84 (10): 39.

Kinnucan, Mark T., Michael J. Nelson, and Bryce L. Allen. 1987. Statistical methods in information science research. *Annual Review of Information Science and Technology* 22: 147-178.

Kinnucan, Mark T., and Dietmar Wolfram. 1990. Direct comparisons of bibliometric models. *Information Processing and Management* 26 (6): 777-790.

Korfhage, Robert R. 1997. *Information storage and retrieval*. New York: John Wiley and Sons.

Kornstädt, Andreas. 1996. SCORE-to-*Humdrum*: A graphical environment for musicological analysis. *Computing in Musicology* 10: 105-130.

Kruhmhansl, Carol, and Jamshed Bharucha. 1986. Psychology of music. In *The new Harvard dictionary of music*, ed. Don Randel, 669-670. Cambridge, MA: Belknap Press.

Lincoln, Harry B. 1989. *The Italian madrigal and related repertories: Indexes to printed collections, 1500-1600*. London: Yale University Press.

Losee, Robert M. 1990. *The science of information.* San Diego: Academic Press.

Marascuilo, Leonard A, and Joel R. Levin. 1983. *Multivariate statitistics in the social sciences: A researchers guide*. Monterey, CA: Brooks Cole Publishing.

McLane, Alexander. 1996. Music as information. *Annual Review of Information Science and Technology* 31: 225-262.

McLean, Bruce Andrew. 1988. The representation of musical scores as data for applications in musical computing. Ph.D. diss., State University of New York.

McNab, Rodger J., Lloyd A. Smith, Ian H. Witten, Clare Henderson, and Sally Jo Cunningham. 1996. Towards the digital music library: Tune retrieval from acoustic input. In *Digital Libraries ' 96, Proceedings of the ACM Digital Libraries conference, Bethesda, Maryland*, 11-18. New York: Association for Computing Machinery.

McNab, Rodger J., Lloyd A. Smith, David Bainbridge, and Ian H. Witten. 1997. The New Zealand Digital Library MELody inDEX. *D-Lib Magazine* (May). Available at: *http://www.dlib.org/dlib/may97/meldex/05witten.html*

Mongeau, Marcel and David Sankoff. 1990. Comparison of musical sequences. *Computers and the Humanities* 24: 161-175.

Nelson, Michael J. 1988. Correlation of term usage and term indexing frequencies. *Information Processing and Management* 24 (5): 541-547.

Nelson, Michael J. 1989. Stochastic models for the distribution of index terms. *Journal of Documentation* 45 (3): 227-237.

Page, Stephen. 1987. Computer tools for music information retrieval. In *Databases in the humanities & social sciences*, ed., Thomas F. Moberg,. Osprey, FL: Paradigm Press.

Page, Stephen Dowland. 1988. Computer tools for music information retrieval. Ph.D. diss., Oxford University.

Palisca, Claude V., ed. 1980. *Norton anthology of western music*, Vol. 1. New York: W.W. Norton.

Parsons, Denys. 1975. *The directory of tunes and musical themes*. New York: Spencer Brown.

Pool, Otto Ede. 1996. The *Apollo* project: Software for musical analysis using *DARMS*. *Computers in Musicology* 10: 123-128.

Pope, Stephen Travis. 1992. *MODE* and *SMOKE*. *Computing in Musicology* 8: 130-134.

Prather, Ronald E. and R. Stephen Elliot. 1988. SML: A structured musical language. *Computers and the Humanities* 24: 137-151.

Prechelt, Lutz and Rainer Typke. 1998. An interface for melody input. Unpublished manuscript.

Randel, Don Michael, ed. 1986. *The new Harvard dictionary of music*. Cambridge, MA: Belknap Press.

RISM. 1997. *Répertoire international des sources musicales: International inventory of musical sources. Series A/II, Music manuscripts after 1600*. CD-ROM database. Munich: K. G. Saur Verlag.

Rubenstein, William Bradley. 1987. Data management of musical information. Ph.D. diss., University of California, Berkeley.

Salton, Gerard. 1975. *A theory of indexing*. Philadelphia: Society for Industrial and Applied Mathematics.

Salton, Gerard. 1989. *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Reading, MA: Addison-Wesley.

Salton, Gerard, and Christopher Buckley. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management* 24 (5): 513-523.

Salton, Gerard, and Michael J. McGill. 1983. *Introduction to modern information retrieval.* New York: McGraw-Hill.

Salton, G., and C.S.Yang. 1973. On the specification of term values in automatic indexing. *Journal of Documentation* 29 (4): 351-372.

Sampson, W.B., and A. Bendell. 1985. Rank order distributions and secondary key indexing. *The Computer Journal* 28 (3): 309-312.

Schaffrath, Helmut. 1992a. The *ESAC* databases and *MAPPET* software. *Computing in Musicology* 8: 66.

Schaffrath, Helmut. 1992b. The retrieval of monophonic melodies and their variants: Concepts and strategies in computer-aided analysis. In *Computer models and representations of music*, ed. Alan Marsden and Anthony Pople. London: Academic Press.

Selfridge-Field, Eleanor. 1994. The *MuseData* universe: A system of musical information. *Computing in Musicology* 9: 11-30.

Selfridge-Field, Eleanor. 1998. Conceptual and representational issues in melodic comparison. *Computing in Musicology* 11: 3-64.

Shannon, Claude E. 1951. Prediction and entropy of printed English. *Bell System Technical Journal* 30: 50-65.

Shannon, Claude E., and Warren Weaver. 1949. *The mathematical theory of communciation*. Urbana, IL: University of Illinois Press.

Snyder, Kerala J. 1992. Cataloguing Sweden's Duben collection. *Computing in Musicology* 8: 27-28.

Sparck-Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28 (1): 11-20.

Sparck-Jones, K. 1974. Automatic indexing. *Journal of Documentation* 30 (4): 393-432.

Srinivasan, Padmini. 1992. Thesaurus construction. In *Information retrieval: Data structures & algorithms*, ed. William B. Frakes and Ricardo Baeza-Yates, 161-218. Englewood Cliffs: Prentice Hall.

Sutton, Joel Brett. 1988. MIRA: A PROLOG-based system for musical information retrieval and analysis. MSLS thesis, University of North Carolina.

Tague, Jean. 1988. What's the use of bibliometrics? In *Informetrics 87/88*, ed. L. Egghe and R. Rousseau, 271-278. Amsterdam: Elsevier Science Publishers.

Tague, Jean. 1990. Ranks and sizes: Some complementarities and contrasts. *Journal of Documentation* 16: 29-36.

Tague-Sutcliffe, Jean. 1992. The pragmatics of information retrieval experimentation, revisited. *Information Processing and Management* 28 (4): 467-490.

Tague, Jean and Paul Nicholls. 1987. The maximum value of a Zipf size variable: Sampling properties and relationship to other parameters. *Information Processing and Management* 23 (3): 155-170.

Temperley, Nicholas. 1993. The problem of definitive identification in the indexing of hymn tunes. In *Foundations of music bibliography*, ed. Richard D. Green, 227-239. Binghamton, NY: Haworth Press.

Tonta, Yasar. 1992. Analysis of search failures in document retrieval systems: A review. *Public-Access Computer Systems Review* 3 (2): 4-53.

Uitenbogerd, Alexandra, and Justin Zobel. 1999. *Matching algorithms for large music databases*. A technical report. Melbourne, Australia: Department of Computer Science, RMIT University.

Ukkonen, E. 1985. Finding approximate patterns in strings. *Journal of algorithms* 6: 132-137.

United Church of Canada. 1930. *The hymnary*. Toronto: United Church Publishing House.

Wolfram, Dietmar. 1992a. Applying informetric characteristics of databases to IR system file design, Part I. *Information Processing and Management* 28 (1): 121-133.

Wolfram, Dietmar. 1992b. Applying informetric characteristics of databases to IR system file design, Part II. *Information Processing and Management* 28 (1): 135-151.

Wolfram, D., C. M. Chu, and X. Lu. 1990. Growth of knowledge: Bibliometric analysis using online database data. In *Informetrics 89/90*, ed. L. Egghe and R. Rousseau, 355-372. Amsterdam: Elsevier Science Publishers.

Yavuz, D. 1974. Zipf's law and entropy. *IEEE Transactions on Information Theory* IT-20 (5): 650.

Zipf, George K. 1935. *Psycho-biology of language.* Boston: Houghton-Mifflin.

Zipf, George K. 1949. *Human behaviour and the principle of least effort.* Reading, MA: Addison-Wesley.

Vita

NAME:	J. Stephen Downie
PLACE OF BIRTH:	Sarnia, Ontario
YEAR OF BIRTH:	1964
POST-SECONDARY EDUCATION AND DEGREES:	The University of Western Ontario London, Ontario 1984-1988 B.A. (Music: Theory and Composition) The University of Western Ontario London, Ontario 1992-1993 M.L.I.S The University of Western Ontario London, Ontario 1993-1999 Ph.D.
HONOURS AND AWARDS:	Leonard Foundation Tuition Scholarship 1984-1986 UWO Graduate Studies Entrance Scholarship 1993-1995 IBM Center for Advanced Studies Research Grant 1994 Best Doctoral Poster, Association for Library and Information Science Education 1996 Best Doctoral Student Paper, Canadian Association for Information Science 1998 Best Doctoral Poster, Association for Library and Information Science Education 1999

RELATED WORK EXPERIENCE:	Lecturer University of Western Ontario 1995-1998
	Lecturer University of Illinois at Urbana-Champaign 1998-
PUBLICATIONS:	See Bibliography