HKU CSIS Tech Report TR-99-05

A study on musical features for melody databases

葉志立Yip Chi Lap Ben Kao
clyip@csis.hku.hk kao@csis.hku.hk
Department of Computer Science and Information Systems,
The University of Hong Kong.

Abstract

Music has an auditory and temporal nature. The same piece can be interpreted in multiple, and often unrelated ways. Together with the limitations of its representations, the design of content-based music retrieval systems on the Web is a challenge. Most literatures on the problem map the problem to existing information retrieval paradigms, mainly that of text, by modelling music as a sequence of features [Lin77][MS90]. However, this mapping raises issues to be solved.

Through the study of the statistical properties of six features including Profile, Note Duration Ratio Sequence, Interval Sequence and their variants, we answer four important questions that arises from mapping. These include the number of musical "alphabets" and "words", whether Zipf's law holds for musical features, whether there are any musical "stopwords", and the range of n for n-gram based music indices. The answers to these questions will affect crucial parameters in music retrieval systems, and whether the mapping and techniques used in text retrieval is effective for music. They will also affect the way to reduce the sizes of musical indices, which we will show is important for music retrieval systems.

Keywords: Information retrieval, Music database, Content-based retrieval, Indexing, Statistical analysis, World-Wide-Web

1 Introduction

The widespread use of multimedia on the Web brings about a number of changes. First, the volume of available media data increases tremendously. Every Web site wants to attract visitors by the use of images, animation, music, or video. Second, the reuse of media data becomes easy. In the past, images, video, animation or music require professionals to produce and special programs or equipments to play. Now, online archives make it easy to use and share these media data on the Web. Even an amateur can design a homepage rich with media data by reusing what is already on the Web. Third, the direction of homepage design shifts from mere information-provision to information-presentation. The Web has become another presentation medium. Information is no longer confined to textual descriptions; it is not difficult to find homepages with background music that use photos and animations to convey information. All these changes mean one thing: media data management becomes increasingly important. Archival, indexing and retrieval of media data are needed to handle large amounts of media data so that reuse is possible for the production of media data-rich presentation forms. Without proper media data management, the availability of the huge amounts of media data is useless; even if one knows that the data is out there somewhere, one cannot reuse what is available for new productions simply because one cannot find it.

Researchers on digital libraries are aware of the problem of media data management. In recent years, there has been an increase in the number of researches on the development of systems that handle images [Wil96], digital video [CKM$^+$95][WKSS96], geographical information [Smi96], or sound effects [KBWW95][WBKW96]. Yet, only a few considered specifically the retrieval of music. Among the relevant ones, few considered the unique characteristics of music in the design of retrieval systems.

Suppose we want to build a content-based music retrieval system on the Web. What is so special about music? What are the major approaches used in the literatures? What are the issues we should look at if we adopt these approaches? These are the questions we are going to solve in the following discussions.

First, what is so special about music? One distinguishing characteristics of music is that it is auditory. Also, by nature, it is temporal. Musical expression depends heavily on the temporal relationship between musical elements. **Rhythmic patterns**[1], **chords** and **broken chords**, **arpeggios** and musical ornaments such as **grace notes** and **trills** all depend on the temporal relationships between elements of music. In contrast, text, image, animation and video are all visual media, and only the latter two are temporal. Because of the visual nature of these media, their corresponding retrieval systems can present the data by putting multiple pieces of them on the same screen. Hence, text excerpts and size-reduced version of images (thumbnail images) are used in many text and image retrieval systems. Even temporal visual data, such as animation and video, can be treated the same way. Each piece of video is time-sampled and size-reduced to produce a series of thumbnail screen shots, or filmstrips. Multiple pieces of temporal visual data can thus be presented this way on the screen without a problem. However, all these techniques cannot be used for music. The auditory nature makes simultaneous playback of multiple pieces of music annoying at best, and the temporal nature would make the playback from the middle of a piece confusing, since there is no musical context. Visualization of music is one possibility; a system can display the score automatically [Rad96], or by the use of specially-designed visualization techniques [MR93][SW97]. However, they often require a level of music literacy or familiarity of the visualization method before they can be used. In other words, the exclusive use of visualization for presentation of music is not suitable for Web-based systems in which users from every walk of life would use. Because of that, it is important for a music retrieval system to match queries accurately, so as to reduce the size of query results. To achieve this, we should first understand how content-based music retrieval systems work, and this brings us to the question of the major approach used in the literatures.

Many literatures on content-based retrieval of music map the problem to existing information retrieval paradigms, predominantly text retrieval, by modelling music as a sequence of features. However, because of its multidimensional nature [SS68] and the limitations of its representations [SF97], no single feature can capture all the essence of music. By multidimensional we mean that the same piece can be simultaneously interpreted in multiple, and often unrelated, ways. For example, the song *Ah! Vous dirais-je maman* (*Twinkle Twinkle Little Star*) is a French folk tune, has an **ABA form**, has a rhythm pattern of 1-1-1-1-1-1-2 repeated six times, and is a children's song, all at the same time. Translated to information retrieval terms, this means multiple indices, or indices with large-sized feature vectors, are needed. Reduction of index size is thus important in music retrieval systems.

By the limitations of music representations, we mean that no representation, digital, written or otherwise, can capture all the aspects of music. Every representation has its strengths and weaknesses. For example, the traditional five-line score can convey the pitch and length of notes of a piece, although features which may modify the volume, pitch and performed length of notes, such as dynamics, tempo, and expression, are often described by the use of words and symbols. These words and symbols, such as "*dolce*" (sweetly) "*vibrato* (rapid periodic fluctuation in pitch), or "*ff*" (Italian *fortissimo*, which means very loud), are subject to the performers' interpretations. Computer representations of music such as Musical Instrument Digital Interface (MIDI) [MID], in contrast, can convey the performed timing, pitch and the volume of each note rather accurately. However, features such as harmony are only implicitly represented.

Indeed, the exact nature of music can only be communicated by its realization. In face of that, and given that we model music as a sequence of features, the selection of features becomes a determinant on the accuracy and effectiveness of music retrieval systems. This brings us to

[1] Section 11 contains a glossary of terms shown in boldface

our third question, that is, what are the issues we should look at, if we model music as a sequence of features?

In music, patterns are abound. Hence, to study the features, we should study their patterns. In the real world, these patterns are often described by ordinary languages. For example, "the chord is a second-inverted G major", "the piece is in common meter", or "it is a waltz". No complete formalism has been established to describe these patterns precisely; even transcribed music leaves room for the performers' interpretations. Different kinds of patterns may exist for different kinds of features. We choose to investigate the statistical properties of some commonly-used musical features to see if they fit the patterns for traditional information retrieval paradigms. In particular, we are going to answer four questions on the commonly-used features of Profile, Note Duration Ratio Sequence, Interval Sequence and their variants, to be introduced in Section 2. A set of pop songs obtained from the Web, which is described in Section 3, was used for our experiments.

Our first question is on the number of feature "alphabets" and possible musical "words". This is an important question because data structures such as suffix tree [McC76][YK98b][YK98a] or n-grams [NTJ$^+$96][LA96] are used for indexing in information retrieval systems. The number of "alphabets" will affect, for example, the maximum branching factor a suffix tree or the maximum n-gram table size. We will investigate this problem in Section 4.

Our second question concerns whether Zipf's law holds for musical features. Zipf's law states that the occurrence probability of words is inversely proportional to its rank in descending occurrence frequency order [Zip65]. Because Zipf's law holds for text, together with the conjecture that the "resolving power" of terms peaks at the middle-frequency range [SM83], techniques such as ignoring frequently-occurring terms can be used to reduce index sizes. Whether this is possible for musical feature sequences will be studied in Section 5.

Our third question asks whether there are any musical "stopwords". In text retrieval, each word read from a document is filtered by a list of "stopwords". Stopwords are words that appear too often in too many documents. This makes them worthless of being included in indices. Since music has a multidimensional nature and often more than one feature is indexed, the analogous musical "stopwords", which reduces the size of index for each feature, would be tremendously useful. We will investigate this question in Section 6.

Our last question is on the value of n for n-gram indices of musical features. A sequence of musical features is similar to a document of Chinese, Japanese or Korean languages: there are no "word boundaries". Since the technique of n-gram indexing is often used for the retrieval of text in these languages, it is possible to use n-grams to index musical features. However, as in the case for these languages, we should select an n. This problem will be studied in Section 7.

After investigating all the four questions by statistical analysis, a discussion and future works section is given in Section 8, and the paper ends by a summary of the results in Section 9.

Although the study on computerized melody database appeared as early as 1977 [Lin77] and there are a few papers on the topic [Sch92][MSW$^+$96] [CCL96] or its variants [KMT93][GLCS95], there has not been much studies on the statistical properties of musical features from an information retrieval point of view. Those on a similar topic, such as [Bar79], are often strongly oriented to audience who are musicians. Through statistical analysis of musical features, this paper hopes to seed researches to fill this gap.

2 Features

Because of the number of dimensions a piece of music can be analyzed (e.g., melody, harmony, rhythm, form, dynamics), it is only possible to introduce some features useful for music retrieval in this paper. Hence, as in most music retrieval literatures, we limit ourselves to features pertaining to the monophonic melodic lines of the pieces, and choose the most commonly used ones, namely melodic Profile, Note Duration Ratio Sequence, Interval Sequence, and its variants "Coarse Interval Sequence" for analysis. Some other features the authors are currently

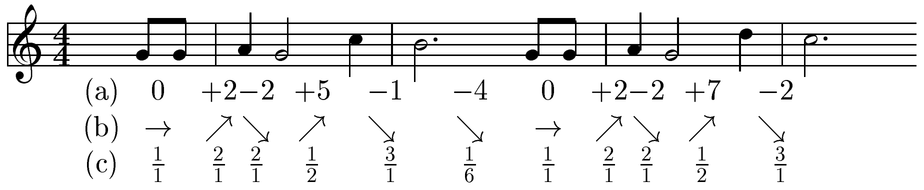

(a) 0 +2 −2 +5 −1 −4 0 +2 −2 +7 −2
(b) → ↗ ↘ ↗ ↘ ↘ → ↗ ↘ ↗ ↘
(c) 1/1 2/1 2/1 1/2 3/1 1/6 1/1 2/1 2/1 1/2 3/1

Figure 1: Excerpt of “Happy Birthday”
(a) Interval Sequence, (b) Profile, (c) Note Duration Ratio Sequence

investigating whose results cannot be included in this paper are also introduced in Section 8.

2.1 Interval Sequence

An Interval Sequence is a sequence of numbers each denoting the musical interval size, in number of semitones, between two temporally consecutive notes. In **equal temperament tuning system**, commonly used in modern Western music, an octave is divided into 12 semitones so that the frequency ratio between notes that are a semitone away is the same. Hence, the interval between C and the C an octave higher is +12, that between C and G is +7, and that between F and E is −1, etc.. An example of the interval sequence for an excerpt of the “Happy Birthday” tune is shown in Figure 1.

Interval sequence is commonly used in past literature because it can be easily computed from computer representations of music, such as the abc musical notation language [abc] or MIDI [MID]. Also, it is independent of the key of the piece. In other words, the musical operation of **transposition** would not affect the interval sequence of a piece. This characteristic makes the support of key-independent queries much easier in music retrieval systems.

2.2 Profile

By the law of trichotomy, each note of a piece can only be of higher, lower, or the same pitch as another note temporally preceding it. Those ups and downs in pitch thus form the piece’s profile. In the written form, those ups, stays and downs are respectively represented as ↗, →, and ↘, “+”, “0” and “−” (e.g., in [Dow78]), or “U”, “R” (for “repeat”) and “D” (e.g., in [The81]). The Profile of an excerpt of “Happy Birthday” is shown in Figure 1.

The study of Profile has a long history, and it has been shown that it relates to the way human memorizes melodies [Dow78]. Some “fake books”, such as [The81], which contain scores of melody and chord names, use the Profile to index on its collection.

2.3 Variants of Interval Sequence

Observant readers may notice that Profile is actually quantized Interval Sequence. It can be obtained by applying the sign function to each element of an Interval Sequence. Indeed, there is more than one way to quantize an Interval Sequence and we can produce a series of “coarse interval sequences” by quantizing it in different ways. For example, putting the interval of zero semitone (unison) at the center, we can group every three intervals to one according to the scheme shown in the row labelled “CI3” (for “Coarse Interval, mapping three intervals to one”) in Figure 2. Coarse intervals CI5 and CI7 can be created in a similar way. Mathematically, an interval of x is mapped to $sgn(x)\left\lfloor\frac{|x|+n}{2n+1}\right\rfloor$ for the feature CI$\{2n+1\}$.

Because of the quantization, the range of coarse intervals will be smaller than that of Interval Sequence. That potentially saves space of indices. Also, the grouping of multiple intervals to one makes the feature more immune to small errors in interval sizes. This can be useful for systems where interval sizes cannot always be found precisely, for example, those derived from human humming inputs, such as [KMT93] or [GLCS95]. The features CI3, CI5 and CI7 are investigated in this paper.

<table>
<tr><td>CI3</td><td>···</td><td colspan="3">-2</td><td colspan="3">-1</td><td colspan="3">0</td><td colspan="3">+1</td><td colspan="3">+2</td><td>···</td></tr>
<tr><td>Interval</td><td>···</td><td>-7</td><td>-6</td><td>-5</td><td>-4</td><td>-3</td><td>-2</td><td>-1</td><td>0</td><td>+1</td><td>+2</td><td>+3</td><td>+4</td><td>+5</td><td>+6</td><td>+7</td><td>···</td></tr>
<tr><td>CI5</td><td>···</td><td colspan="5">-1</td><td colspan="5">0</td><td colspan="5">+1</td><td>···</td></tr>
</table>

Figure 2: Interval — coarse interval mapping

2.4 Note duration ratio sequence

Timing information is important in dealing with rhythm. One of the most convenient, and thus most commonly used, timing parameters is the ratio of sounding durations between two temporally consecutive notes. A Note Duration Ratio Sequence is thus a sequence of such ratios for each temporally consecutive note pair. Figure 1 shows an example Note Duration Ratio Sequence. The first $\frac{1}{1}$ means that the duration of the first and the second note are the same, and the first $\frac{2}{1}$ indicates that the third note has twice the length of the second note.

In some literatures, this sequence is simply called "rhythm". This is a misnomer because other characteristics of rhythm, such as accent, are ignored in the construction of Note Duration Ratio Sequence.

Similar to Interval Sequence which deals with relative note pitch (i.e., interval), Note Duration Ratio Sequence deals with relative note timing. So, just as interval sequence is invariant to transposition (frequency scaling), Note Duration Ratio Sequence is invariant to time-scaling. The Note Duration Ratio Sequence for a piece is the same regardless of the speed it is performed.

In real performances, notes are not always performed in the exact length as notated. The reasons are twofold: delibrate departure from the notation for expressive purposes, and limitations of human timekeeping and motor systems. Hence, finding Note Duration Ratio Sequence by simple note duration extraction and division works only for a few nicely preprocessed music files. For a Web-based music retrieval system which has no control on how music is preprocessed, quantization is needed. Onset quantization, a method that maps the onset times of notes to points in a fixed grid at some time resolution, is perhaps the most commonly implemented quantization method because of its simplicity. For more information on the quantization problem and the approaches to solve it, readers are referred to [DH92] and [DHdR].

3 Music collection

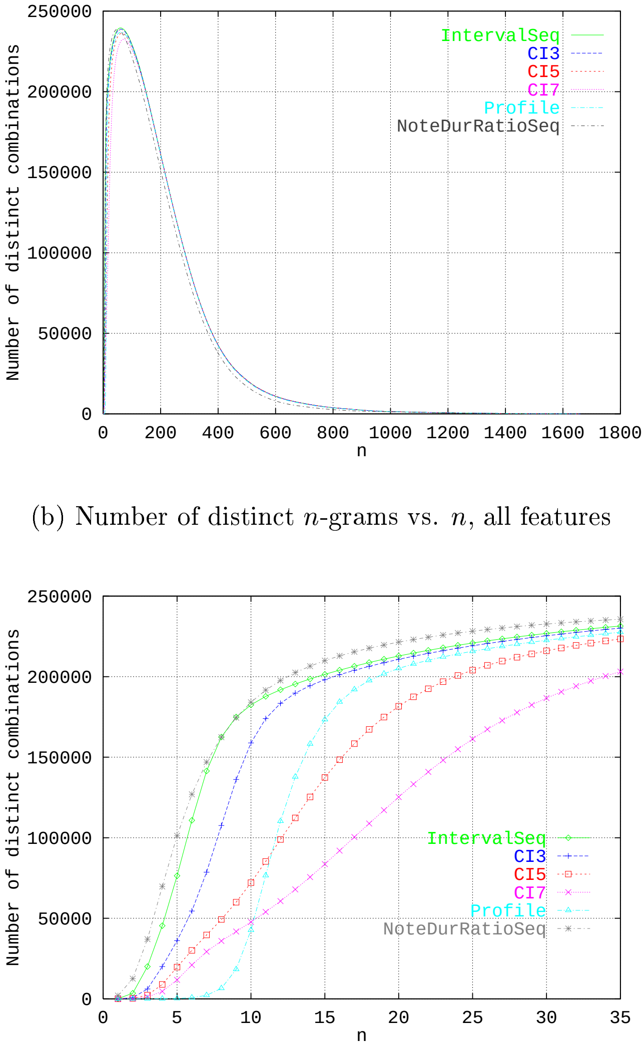
A music collection is needed before any statistical analysis can be done. Our music collection consists of 1003 files of songs and music popular in Hong Kong, Taiwan, or Japan, all as MIDI files. As opposed to most literature on music retrieval which use sets of well-quantized and well-tagged folk tunes or classical pieces for experiments (one notable work includes [MSW$^+$96]), we obtained these MIDI files from the Web. This not only makes our collection more up-to-date, but also parallels the situation of our target music retrieval application: content-based retrieval of music on the Web. The dynamic nature of the Web makes frequent update of indices necessary. Since popular songs are always on the vogue, many new files that need to be processed by a music retrival system would be that of popular music.

Since we deal with monophonic features only, MIDI tracks containing melodies have been manually extracted for the experiments. Some of these extracted tracks contain the introduction sections of the songs besides the main melody. Automatic onset quantization is done on the melodies before feature extaction. The quantization level is automatically derived from the time signature and other parameters of the piece, so that sections of a song with different time signatures can be quantized differently. Six features, namely Interval Sequence, CI3, CI5, CI7, Profile and Note Duration Ratio Sequence, are used for analysis.

Len n	nComb theory a	nComb actual b	Count of n-grams k	Value of k/b
1	255^1	117	357657	3056.89
2	255^2	3484	356654	102.36
3	255^3	20073	355651	17.71
4	255^4	45417	354648	7.80
5	255^5	76399	353645	4.62
6	255^6	110828	352642	3.18
7	255^7	141424	351639	2.48
8	255^8	162531	350636	2.15
9	255^9	174979	349633	1.99
10	255^{10}	182460	348630	1.91
11	255^{11}	187729	347627	1.85
12	255^{12}	191875	346624	1.80
13	255^{13}	195409	345621	1.76
14	255^{14}	198553	344618	1.73
15	255^{15}	201430	343615	1.70
16	255^{16}	204050	342612	1.67
17	255^{17}	206478	341609	1.65
18	255^{18}	208747	340606	1.63
19	255^{19}	210857	339603	1.61
20	255^{20}	212815	338600	1.59
21	255^{21}	214632	337597	1.57

a: Theoretical number of combinations
b: Actual number of distinct combinations
k: Total number of n-grams, distinct or not

(a) Interval Sequence statistics



(b) Number of distinct n-grams vs. n, all features

(c) Curves zoomed in for $n \leq 35$

Figure 3: Feature statistics

4 Musical "alphabets" and "words" — how many?

The number of "alphabets" for music features affect a number of parameters in music retrival systems. These include the maximum branching factor of suffix tree indices [YK98b][YK98a], and the theoretical number of combinations of n-grams which relates to the maximum number of n-gram entries. For example, in MIDI representation of music, a total of 128 notes are allowed, ranging from C0 (the C five octaves below middle C) to G10 (G above the C five octaves above middle C). Hence, the number of possible interval sizes ranges from -127 to $+127$ semitones, giving a total of 255 combinations. Thus, the possible number of n-grams of intervals would be 255^n, a number that increases exponentially with n. Theoretically, all these combinations are possible in music feature sequences. Fortunately, as described by Birkhoff's Aesthetic Theory [Sch91], music is not like white noise; some combinations happen much more likely than others.

Figure 3(a) gives the number of length-n feature sequences, or n-grams, on Interval Sequence for the 1003 melodies of our music collection. There are a total of 357657 intervals in our collection, as can be read from the count of n-gram column for $n = 1$. These intervals fall into only 117, that is, about 45.9 percent, of all the 255 possible interval sizes. A closer examination of the feature files revealed that the range of interval sizes varied only from -74 to $+72$. Indeed, given the human hearing range of 20Hz to 20kHz, which covers a range of about

$12 \log_2 20000/20 \approx 120$ semitones, allowing interval sizes from -120 to $+120$ semitones would be more than adequate for Interval Sequences. Interval sizes whose absolute values are too large, such as $+168$ or -255, do not make musical sense. Similarly, the ranges for the features CI3, CI5 and CI7 would be from -40 to $+40$, -24 to $+24$, and -17 to $+17$ respectively. Profile, of course, has only three "alphabets", $\nearrow$, $\searrow$ and $\rightarrow$.

Unlike interval-related features, we do not have a bound on the number of "alphabets" for Note Duration Ratio Sequences, since note lengths can be arbitrary. Indeed, for our collection, there are 2046 note duration ratio combinations. However, these combinations are rather skewed in distribution; the most frequent 24 combinations appear 80 percent of the time, the most frequent 68 combinations appear 90 percent of the time, and the most frequent 146 combinations appear 95 percent of the time. The implication of such a distribution will be discussed in more detail in Section 5.

As n increases, we observe a number of trends. First, the actual number of n-gram combinations (b Table 3) does not increase exponentially as its theoretical number (a in the table). Second, the total count of n-grams decrease linearly. Third, the ratio of the count of n-grams to the actual number of n-gram combinations approaches one.

We can examine our first observation closer by plotting the number of n-gram combinations for each feature (b of Figure 3(a)) against n in Figures 3(b) and 3(c). It is found that the number of n-grams does not increase exponentially for the full range of n. After a short period of exponential-like increase, the number of n-grams reaches its maximum gently at $n \approx 60$ and falls afterwards. At smaller n of around 200 the fall was rather quick; a slope of -700 distinct combinations per unit increase of n was typical. As n increases, the rate of fall gradually slows down to -1 distinct combinations per unit increase of n.

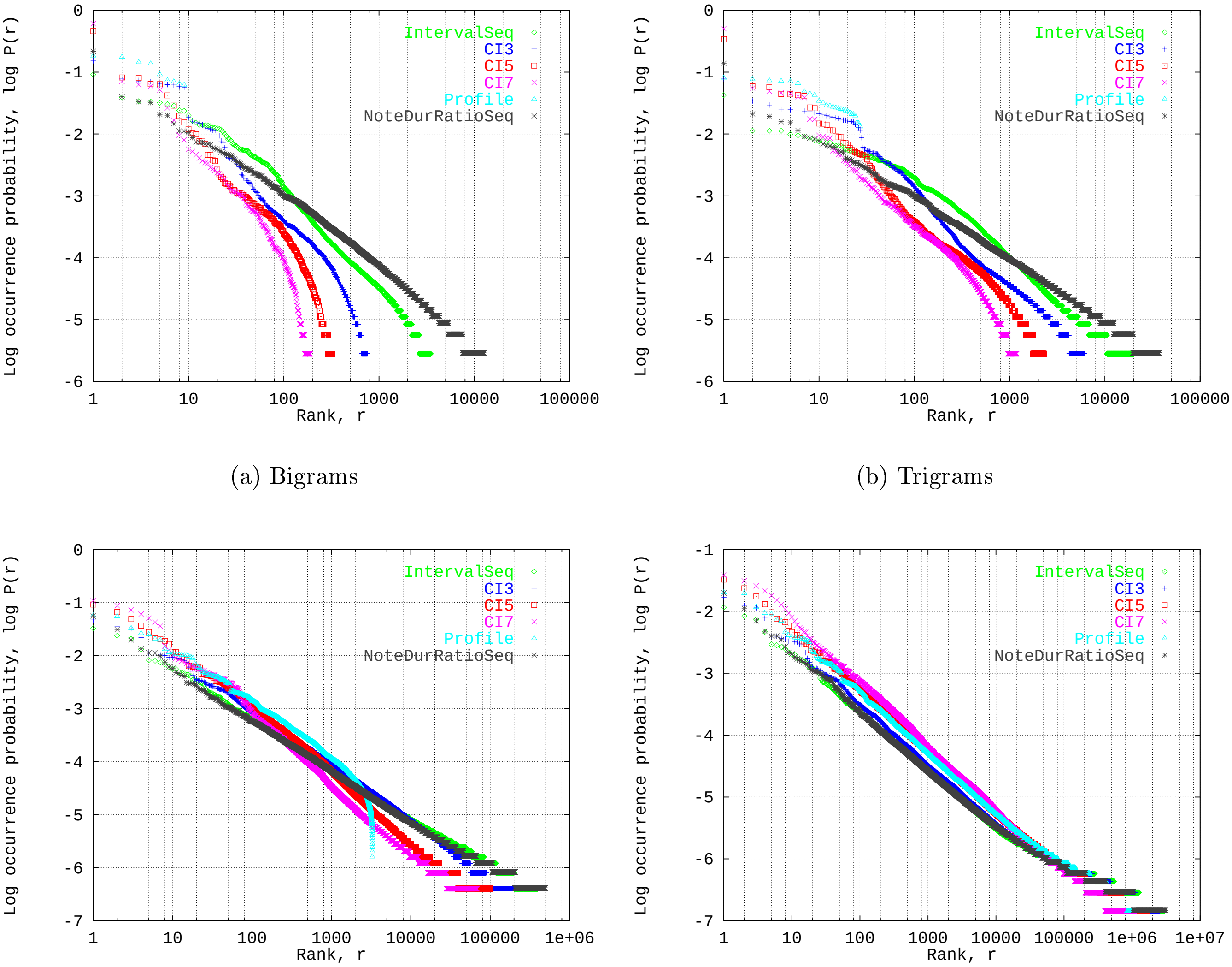
The nonexponential increase indicates that interval sequences in music is not completely independent of each other. The quick fall at around $n = 200$ to 400 shows that the length of feature sequences for quite a number of songs are in that range; an increase of n excludes those songs from forming n-grams and thus cause the drop of n-grams combinations. The more gradual fall afterwards indicates that less and less songs have long feature sequences, and the linear fall at large n implies that very long n-grams are mostly distinct.

Since the increase in the number of n-grams is slower than exponential, n-gram indexing techniques can be used for the features under investigation if there is some form of n-gram table size control. The question on the choice of n will be studied in Section 7.

Our second observation, that the total number of n-grams decreases linearly with n, is a simple consequence of n-gram usage: a sequence of length k has $(k - n + 1)$ n-grams, so an increase of n corresponds to a decrease in $(k - n + 1)$.

Our third observation, that k/b approaches 1 as n increases, means that for any given n-gram, on average, fewer exact matches to the song sequences are possible as n increases. Long n-grams, which are mostly distinct, will almost uniquely identify the section of music. For example, an n of 4 would give 7.8 Interval Sequence pattern matches in our database on average, if every Interval Sequence 4-gram were chosen with equal probability. However, the worst case behaviour depends on the distribution of Interval Sequence 4-grams. We will investigate the n-gram distributions in Section 5 by studying whether they obey Zipf's law.

Figure 3(c) shows an enlarged view of Figure 3(b) for $n \leq 35$. Expectedly, the number of feature combinations of the coarse interval sequences are in general smaller than that for Interval Sequences. However, for n larger than, say, around 25, the numbers become more or less the same. Quantization by grouping more interval sizes together in general makes the increase of number of combinations for small n more gentle. This is in contrast of the feature Profile, where the increase in number of combinations at around $n = 10$ is quick. Indeed, a look at the raw data for Profile reveals that it follows the exponential curve quite closely until around $n = 10$, after which it is actually slowed down and it becomes more similar to other interval sequence-related curves.



(a) Bigrams

(b) Trigrams

(c) Sequence length ≤ 7

(d) Sequence length ≤ 20

Figure 4: log occurrence probability versus rank

5 Does Zipf's law hold for musical features?

One of Zipf's results on natural languages states that the occurence probability of words is inversely proportional to its rank in descending occurrence frequency order [Zip65]. This result is commonly known as "Zipf's law". In text retrieval, it was conjectured that the "resolving power" of terms peaks at the middle occurrence frequency range [SM83]. Because of this, techniques such as ignoring frequently-occuring terms can be used to reduce index sizes very effectively, as most frequently-occurring terms that obey Zipf's law constitute a major proportion of term occurrence probability. The same technique can be used effectively if Zipf's law holds for musical features.

To investigate whether this is the case, we count the occurrence frequency of all feature sequences whose lengths are within a certain range, rank them according to their occurence frequency, and plot the graphs of the logarithm of their relative occurence probability, $\log_{10} P(r)$, against the rank r in log scale. Data obeying Zipf's law should form a straight line on the graph. Four feature sequence length ranges are considered: those of length 2 (bigrams), length 3 (trigrams), those of length no more than 7, and those no more than 20.

Figure 4 shows the four graphs. It can be seen that for bigrams and trigrams, the curves do not approximate straight lines very well. They convex upwards. This convex behaviour means that the occurrence probabilities fall slower than expected when r is small but faster when r

Rank	IntervalSeq	CI3	CI5	CI7	Profile	NoteDurRatioSeq
1	0	0	0	0	$\searrow$	$\frac{1}{1}$
2	-2	-1	0 0	0 0	$\nearrow$	$\frac{1}{1}\ \frac{1}{1}$
3	+2	+1	0 0 0	0 0 0	$\rightarrow$	$\frac{1}{1}\ \frac{1}{1}\ \frac{1}{1}$
4	0 0	0 0	0 0 0 0	0 0 0 0	$\searrow\nearrow$	$\frac{1}{1}\ \frac{1}{1}\ \frac{1}{1}\ \frac{1}{1}$
5	-3	0 -1	0 0 0 0 0	0 0 0 0 0	$\nearrow\searrow$	$\frac{2}{1}$
6	+3	0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	$\searrow\searrow$	$\frac{1}{2}$
7	-1	-1 +1	+1	0^{7}	$\nearrow\nearrow$	$(\frac{1}{1})^{5}$
8	+1	-1 -1	-1	0^{8}	$\rightarrow\rightarrow$	$(\frac{1}{1})^{6}$
9	0 0 0	+1 -1	0^{7}	0^{9}	$\searrow\nearrow\searrow$	$(\frac{1}{1})^{7}$
10	-2 +2	+2	0^{8}	0^{10}	$\nearrow\searrow\nearrow$	$\frac{1}{1}\ \frac{2}{1}$
11	+5	+1 0	0 -1	0^{11}	$\rightarrow\searrow$	$(\frac{1}{1})^{8}$
12	-2 -2	-1 0	+1 0	0^{12}	$\searrow\searrow\nearrow$	$\frac{2}{1}\ \frac{1}{2}$
13	+2 -2	+1 +1	0^{9}	+1	$\nearrow\searrow\searrow$	$\frac{1}{2}\ \frac{1}{1}$
14	0 -2	0 +1	0 +1	0^{13}	$\searrow\nearrow\nearrow$	$(\frac{1}{1})^{9}$
15	-2 0	-2	-1 0	0^{14}	$\nearrow\rightarrow$	$\frac{3}{1}$

Table 1: Most frequent sequences. x^n means x repeated n times.

is large. Interestingly, this convex behaviour is also found in the distribution of alphabets (not words) for natural languages such as English, Chinese and Hebrew, as well as those of amino acids, which are building blocks of proteins [Sht94]. Bigrams and trigrams thus seem to be more alphabet-like than word-like.

In Figure 4(c), the Profile curve still shows convex behaviour, while all other curves approximate straight lines well. Figure 4(d), which includes the statistics for feature sequences of length no more than 20, shows that all features obey Zipf's law. Thus, music feature sources can be modelled as stochastic processes [Zip65]. Consequently, techniques such as co-occurrence analysis can be applied for feature analysis, clustering and indexing. Profile, because of its small alphabet size, require longer length to make them behave statistically like musical "words". That is, the "word length" for Profile is expected to be larger than that of other features. Whether this is the case will be investigated in Section 7.

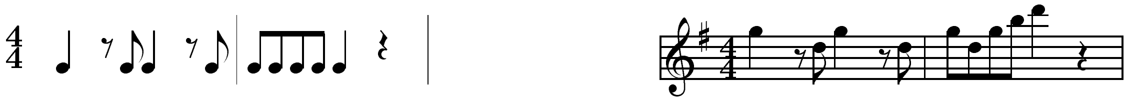
6 Are there any musical "stopwords"?

In keyword-based text retrieval systems, words are extracted from a document and then "weighted" to determine their significance as keywords. Before the weighting, words that appear too often in too many documents, such as "the" or "is", are filtered out. Such words are called stopwords.

Musical features like Interval Sequence do not have "word boundaries". However, the question of whether there are any "stopwords" is still significant. By knowing those stopwords, or frequently-occurring sequences, for example, music analysis programs can construct heuristics for faster analysis. Also, music retrieval systems can use them for query refinement, and index sizes can be reduced by excluding them.

Table 1 shows the most frequent sequences for the features under consideration. Single small intervals (-3 to $+3$ semitone) are common in our melody collection, as well as sequences of small intervals (e.g., -2 -2) or repeats. Of particular interest is that **perfect fourth** upleaps (interval size of $+5$ semitones) is rather commonly used in our melody collection. A closer look at the raw data shows that perfect fourth downleaps (-5 semitone intervals) and **perfect fifth** intervals (± 7 semitones) are among the top 25 as well.

For coarse interval sequence CI3, zero or size one jumps, which correspond to intervals between -4 to $+4$ semitones, are most common. For CI5 and CI7, consecutive sequences of zeros or short sequences of zeros or small jumps are most frequent. These indicate that intervals



(a) Note duration only

(b) Score

Figure 5: Excerpt of Mozart's K.525 *Eine Kleine Nachtmusik*

with large magnitudes are less commonly used, and thus give more "surprises".

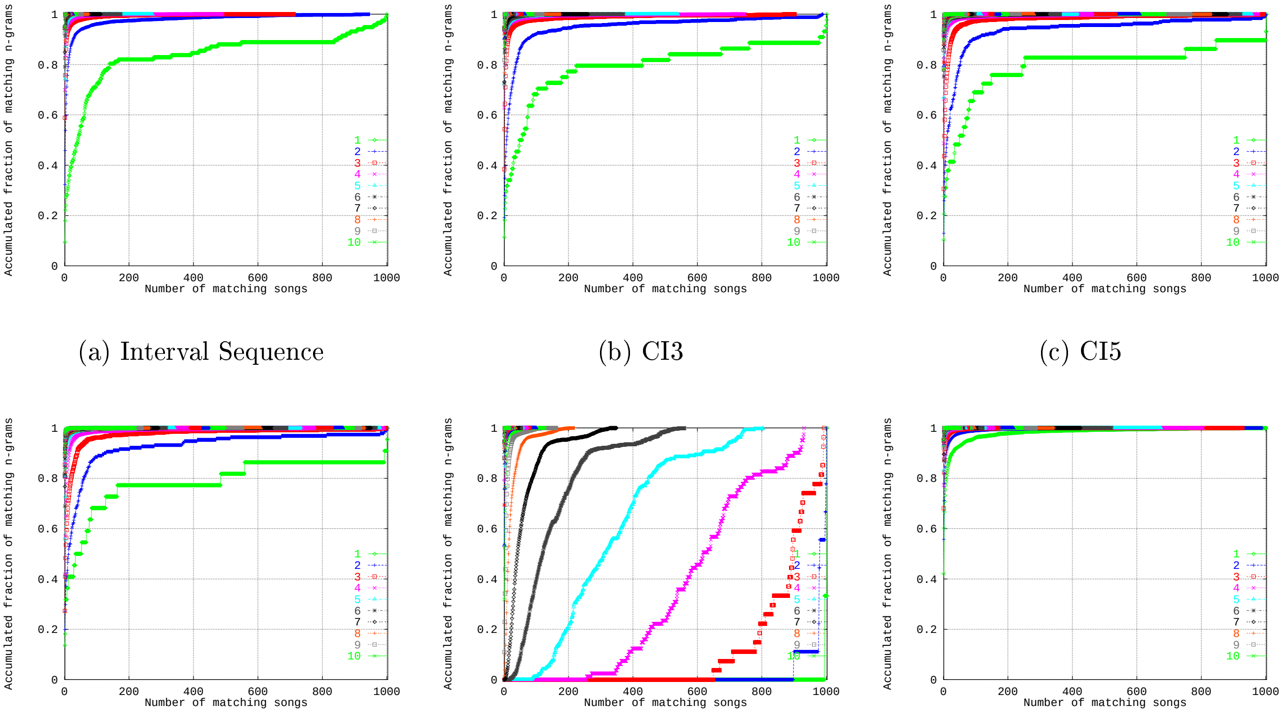
If we consider interval unigrams alone, the most frequently occurring ones, in descending order of their occurrence frequency, are 0, −2, +2, −3, +3, −1, +1, +5, −5, +7, +4, −4, −7, +9, +12, −9, −12, +8, and −8. Thus, one- or two-note intervals in a scale, which correspond to interval sizes of -4 to +4, are most commonly used in melodies. This is in line with the observations in [Dow78], where a marking method that uses less memory for small intervals is proposed. Thus, to reduce index size, music retrieval systems that index on Interval Sequence or coarse interval sequences could consider only those sequences with intervals whose magnitude are greater than a certain threshold. Also, since the highest-ranked feature sequences tend to be of length one or two, musical "words" useful for indices should usually be longer. However, consecutive sequences of zeros, especially for CI5 and CI7, are in general poor features.

For Profile, the raw data showed that both ↘ and ↗ appear with more or less the same probability, while → is about sixty percent as frequent as either of them. This phenomenon, that the flat profile → does not appear often, is in contrast with the case for Interval Sequence and coarse interval sequences, where the unison interval appears rather frequently. One of the reasons for this is that the quantization method applied to Interval Sequence for obtaining Profile, as opposed to that for the coarse interval sequences, is nonuniform. All pitch upleaps and downleaps are mapped to ↗ and ↘ respectively, while only the unison interval is mapped to →. Thus, the flat profile becomes an important feature. Indeed, the first Profile feature sequence with nonconsecutive flat profiles ranked 54th in the table, with an occurrence probability of less than 0.04 percent, infrequent enough to be useful index sequences. An example of nonconsecutive flat profile sequence is "→↗→↗→↘" from "*Ah! Vous dirais-je maman*" (*Twinkle Twinkle little star*, d d s s l l s) or Jospeh Haydn's *Andante* theme of *Surprise Symphony* (d d m m s s m).

For Note Duration Ratio Sequence, the predominance of the ratio $\frac{1}{1}$ as components of the sequences implies that pieces in which every note is of the same length do not distinguish themselves from others. In other words, change of note duration carries distinguishing information of musical pieces. Interestingly, both examples we have just shown to have distinguishing Profile do not have very distinguishing Note Duration Ratio Sequences. Yet, some pieces, such as Mozart's K.525 *Eine Kleine Nachtmusik* (Figure 5), are especially recognizable even by rhythm only. The ten most frequently-occurring note duration ratio unigrams, in decreasing order of frequency, are $\frac{1}{1}$, $\frac{2}{1}$, $\frac{1}{2}$, $\frac{3}{1}$, $\frac{1}{3}$, $\frac{2}{3}$, $\frac{3}{2}$, $\frac{4}{1}$, $\frac{1}{4}$ and $\frac{1}{6}$, and they are all among the top 35 of all Duration Ratio Sequences whose lengths are no more than 20.

7 What should *n* be for *n*-gram indices?

For natural languages without word separators, such as Chinese, Japanese or Korean, *n*-grams are often used for text indexing. This way, a piece of text can be indexed without invoking computationally expensive or inaccurate parsing routines to find the word boundaries. Yet, we have to choose a value for *n*. As discussed in Section 4, *n* controls the theoretical number of *n*-grams. Since this number is exponential to *n*, a small *n* is preferred. However, sometimes a larger *n* would fit some characteristics of the language better. For example, an *n* of 3 matches the length of people's names in Chinese, while an *n* of 4 matches the length of many Chinese idioms. In practice, *n* is chosen to match the word length characteristics of the langauge. For example, *n* is often chosen as 2 for Chinese since more than two third of Chinese words are formed by two characters [Sue86], and for Korean, *n* is 2 or 3 for the same reason [LA96].



(a) Interval Sequence

(b) CI3

(c) CI5

(d) CI7

(e) Profile

(f) NOMDRRSeq

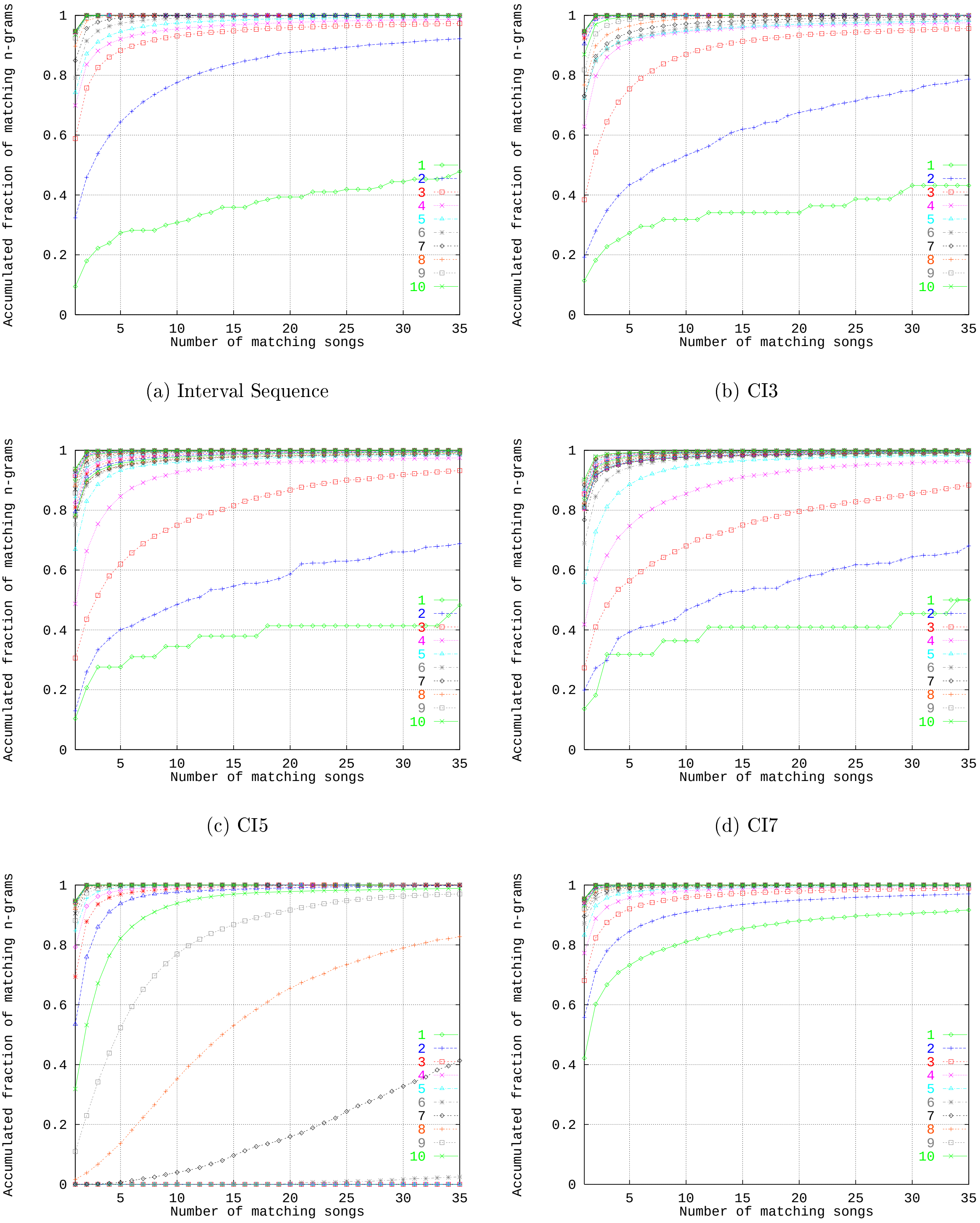
Figure 6: Feature matching graphs

One of the characteristics of music feature sequence is remarkably alike with those of natural languages: there is no "word separator" in a music feature sequence unless some computationally expensive or inaccurate procedures, such as music segmentation and phrase analysis, has been done. Given that n-gram indexing works for those natural languages, we can use the technique for indexing music. So, the problem of the choice of n arises. To study this, we performed pattern matching experiments on our music collection.

Given an n, we want to know the number of songs that the n-grams can match. Our goal is to plot a graph that shows the fraction of all n-grams that appeared in the database which matches no more than a certain number of songs on the x-axis. Since every n-gram for a fixed n may match a different number of songs, to plot the curves, we do the following for every feature being studied. First, we fix an n. Then, we enumerate all the n-grams in our database and match each of them with our collection of 1003 songs. This way, we know the number of songs that match each n-gram. When all n-grams are matched, we invert the statistics and obtain the number of n-grams that match a given number of songs. With that, we can know the total number of n-grams, and thus the fraction of all n-grams in the collection, that matches no more than a given number of songs. Since in practice n would not be very large, and the number of distinct n-grams peaks at $n \approx 60$ for our collection, we repeated the experiments for every $n < 30$.

Figure 6 shows the graphs for the six features. The lowest (bottom rightmost) curves in the graphs are the ones for $n = 1$, the next lowest for $n = 2$, and so on. From these graphs it is found that for the same n, Note Duration Ratio Sequence matches a smaller number of songs than Profile does in general. Hence, for the same feature sequence length, Note Duration Ratio Sequence is most discriminating while Profile is the least discriminating, if every feature sequence of the same type is selected with the same probability. To study what n should be for a practical system, detailed views of the graphs, for number of matching songs no more than 35, are shown in Figure 7.

Suppose we want to design an n-gram-based music retrieval system that, in 80 percent of the time, retrieves no more than 20 songs when an n-gram in the database is randomly chosen. The minimum n should thus correspond to that for the line above the point $(20, 0.8)$ on



(a) Interval Sequence

(b) CI3

(c) CI5

(d) CI7

(e) Profile

(f) NOMDRRSeq

Figure 7: Feature matching graphs, zoomed in for number of matching song ≤ 35

the graphs. If Interval Sequence is used, we found that n should be no less than 2. For CI3 and CI5, n should be no less than 3, and for CI7 no less than 4. Profile, because of its small alphabet size, requires a longer sequence length of 9 or more, but for Note Duration Ratio Sequence, a length of 1 would do the work. Yet, since it was shown in Section 6 that the most frequent Note Duration Ratio Sequences are all among the top 35 of Duration Ratio Sequences of length no more than 20, a larger n should be chosen and a minimum of 2 is needed.

Note that in this set of experiments, we enumerated all the n-grams for a fixed n. Effectively, it is assumed that every length-n feature sequence would be selected with equal probability; the distribution of n-grams has not been taken into account. This is one of the reasons for the small n found for Note Duration Ratio Sequence.

Because of the skewed distribution of features n-grams, system designers should take the above results as a minimum value for n and be liberal to choose a larger value whenever practical. The authors are carrying out experiments that take this distribution into account. More details about this and other works in progress will be introduced in Section 8.

8 Discussion and future work

In the previous sections, we have investigated the statistical properties of melodic Profile, Interval Sequence, three coarse interval sequences, and Note Duration Ratio Sequence. These are all features suitable for monophonic music sequences. Thus, manual extraction of melody lines was done as a preprocssing step for our experiments. In reality, even simple MIDI files on the Web often contain accompaniments and chords; they are polyphonic. Hence, to handle these features in a practical music retrieval system, we need to extract melody lines automatically or find some polyphonic features. The authors are currently investigating some features, such as chord progressions, that are suitable for handling polyphonic music and their queries.

In our study of musical "alphabets" and "stopwords", we used the frequency occurrence statistics. Experiments that elaborate the results by using more information retrieval notions as measures, such as "term frequency × inverse document frequency" are also being done. Also, experiments that take distribution of n-grams into account to find n, discussed in Section 7, are under way.

Literatures that treat music as a sequence of features often suggest using traditional string or multidimensional matching methods to handle inexact or nearest-neighbour queries (e.g., [CCL96]). These methods often require a metric that specifies how alike two given features are. In music, there is often multiple such distance measures for any given type of feature, and they are often not metrics. For example, for a given key in the major mode, the **dominant** is often considered as "nearest" to the **tonic**, since they are a perfect fifth away, and the notes are thus consonant. However, two notes two perfect-fifths away are considered to be "further away" than "twice the distance" of notes a perfect fifth away. This is because the combined interval, an octave-and-a-second, is dissonant. Few literatures on music databases take these domain-specific characteristics into account; most of them simply assume that there is some magical way to fit music into the traditional information retrieval framework. The authors are currently investigating how different distance measures and other music-specific characteristics would affect the performance of music retrieval systems. For more information on musical "distances", readers are referred to [Tay93], [Tay92], and [MS90].

In this paper, we have only reported statistics of features by themselves. The authors have collected joint-feature statistics and done some other experiments, such as forming queries that start and end on the beat by sampling melodies in the database. Unfortunately, the results of these experiments cannot be reported here because of space limitations.

Experiments in this paper are done using a collection of popular music. While we would like to enlarge our song collection and have a much larger collection of tunes in archives, we did not mix them all, since different genre of music may give different statistical results. It is intersting to investigate whether and how different genre of music affects the effectiveness of music retrival systems. Also, although the mode of music (e.g., major, minor, Dorian, Mixolydian) along with the profile and interval, has been identified to play an imporant role in human memory of monophonic melodies [Dow78], not much emphasis has been put on them in melody database literatures. Further research on how the modes of musical pieces can be used in music retrieval systems is needed.

9 Summary

The auditory and temporal nature of music, together with its multidimensional nature and the limitations of its representations, pose challenges on the design of content-based music retrieval systems on the Web. Since the major approach in the literature is to map music retrieval to other information retrieval paradigms such as those for text, we answer four important questions that arise from the mapping through the study of six features on our collection of 1003 popular songs obtained from the Web. These questions are, first, the number of musical "alphabets" and "words", which will affect crucial parameters in music retrieval systems. Second, whether Zipf's law holds for musical features, which affects whether some text retrieval techniques are effective for music features. Third, whether there is any musical "stopword", which would help to reduce the size of indices, and last, and the range of n for n-gram indices on musical pieces, which affect the maximum table of size n-gram indices. The six features studied include Interval Sequence that encodes pitch information of temporally consecutive notes, three coarse interval sequences CI3, CI5 and CI7 which are Interval Sequences uniformly quantized with different step sizes, Profile which is a nonuniformly quantized Interval Sequence, and Note Duration Ratio Sequence which encodes timing information of a melody.

In Section 4, we found that interval size that ranges from -120 to $+120$ semitones would be enough for musical sequences, although less than half of them were used by all the intervals in our song collection. The corresponding ranges for the features CI3, CI5 and CI7 are from -40 to $+40$, -24 to $+24$, and -17 to $+17$ respectively, and Profile, by definition, has only three "alphabets", $\nearrow$, $\searrow$ and $\rightarrow$. As regards to Note Duration Ratio Sequences, there is no theoretical bound on the number of alphabets, since note lengths can be arbitrary. Yet, statistics on our collection showed that the distribution of Note Duration Ratios is rather skewed, among 2046 combinations, the most frequent 68 combinations appear 90 percent of the time.

It was also found that the number of distinct n-grams does not increase exponentially all the way as n increases. It peaks at $n \approx 60$ and falls afterwards. The number of distinct n-grams for quantized Interval Sequences is smaller than Interval Sequence in general, but at n larger than around 25, the difference shrinks. It was also found that most feature sequences have a length of 200 to 400, and long n-grams are mostly distinct. Also, the average number of sequence match decrease as n increases. For example, at $n = 4$, on average there are only 7.8 pattern matches for the feature Interval Sequence if every Interval Sequence 4-gram were chosen with equal probability.

On the study of Zipf's law conformance in Section 5, it was found that bigrams and trigrams, as well as shorter Profile sequences, show convex behaviour on a graph and behave more alphabet-like than word-like. When both short and long features are considered, all six features obey Zipf's law and thus can be modelled as stochastic processes. This implies that techniques such as co-occurrence analysis can be applied for analysis, clustering and indexing purposes.

In finding musical "stopwords" in Section 6, it was found that small intervals of about -3 to $+3$ semitones are most common in our music collection. In other words, larger intervals give more surprises and indices could thus consider only feature sequences with large magnitude intervals to reduce their sizes. Runs of zeros are especially common in coarsely quantized features such as CI5 or CI7. In contrast, because of nonuniform quantization, the flat Profile $\rightarrow$, which corresponds to the unison interval, is important in Profile feature sequences. The first Profile feature sequence with nonconsecutive flat profiles ranked 54th with an occurrence probability of less than 0.04 percent in our music collection, and is infrequent enough to be a useful index sequence.

Statistics on Note Duration Ratio Sequences show that pieces in which every note is of the same length do not distinguish themselves from others. In other words, change of note duration carries distinguishing information of musical pieces. Expectedly, the most frequently-occurring note duration ratios are simple ones such as $\frac{1}{1}$, $\frac{2}{1}$, $\frac{1}{2}$ or $\frac{3}{1}$.

Result on pattern matching experiments reported in Section 7 showed that in general Profile needs a longer query sequence than other features. If every n-gram combinations are selected with equal chance, and we want to design an n-gram-based music retrieval system that, in 80 percent of the time, retrieves no more than 20 songs, the n for Interval Sequence should be no less than 2. That for CI3 and CI5 should be no less than 3, CI7 no less than 4, and for Profile no less than 9. For Note Duration Ratio Sequence, a practical length of at least 2 is needed.

Because of its auditory, temporal, multidimensional nature, and also the limitations of its representations, music retrieval is a rich area for study. The author is undertaking a number of studies on them. These include polyphonic and joint-feature statistics, the use of more information retrieval notions to elaborate results in this paper, the inclusion of n-gram distribution to find n for n-gram indices, the study on the implications of the use of different musical distance measures, and the formation of queries that start and end on the beat by sampling techniques. Other interesting problems on the subject include automatic melody extraction, a study on the statistical difference of different genres of music, and the use of mode of music on music retrieval systems.

10 Acknowledgements

Special acknowledgements are given to CHENG Man Yee, HO Wai Shing, TANG Fung Michael, and YUEN Kin Wai for their painstaking extraction of melodic lines from the song collection which made the experiments possible.

11 Glossary

A major reference of the musical terms is [Jac91].

ABA form The layout of a piece so that the first and the third sections are the same while the middle is different.

Arpeggio Chord that is performed "spread out"; notes are not sounded simultaneously but in close succession.

Broken chord Notes of a chord that are played one after another, for example, one note on each beat.

Chord Simultaneous combination of (usually not less than three) notes.

Dominant The fifth note on a scale, in relation to the key note.

Equal temperament tuning system The "tempering" (slight lessening or enlarging) of musical intervals away from the natural scale deducible by physical laws such that each semitone is made an equal interval; notes that are a semitone away has fixed frequency ratio.

Grace note Musical decorations in which one note (the grace note) in played quickly before another note.

Perfect fifth An interval of five semitones.

Perfect fourth An interval of seven semitones.

Rhythmic pattern The pattern of music concerned with the distribution of notes in time and their accentuation.

Tonic The first degree, or the key note, of a scale.

Trill A musical ornament, also called "shake", consisting of rapid alternation of the written note and note above.

Transposition Transferring (write down or perform) music at a pitch other than the original.

References

[abc] http://www.gre.ac.uk/~c.walshaw/abc/index.html. Homepage.

[Bar79] John Barnes. Bach's keyboard temperament: Internal evidence from the well-tempered clavier. *Early Music*, 7(2):236–249, April 1979.

[CCL96] Ta-Chun Chou, Arbee L.P. Chen, and Chih-Chin Lu. Music databases: Indexing techniques and implementation. In *Procedings IEEE International Workshop on Multimedia Data Base Management Systems 1996*, 1996.

[CKM+95] M. Christel, T. Kanade, M. Mauldin, R. Reddy, and M. Sirbu. Informedia digital video library. *Communications of the ACM*, 38(4):57–58, April 1995.

[DH92] Peter Desain and Henkjan Honing. The quantization problem: Traditional and connectionist approaches. In Mira Balaban, Kemal Ebcioğlu, and Otto Laske, editors, *Understanding Music with AI: Perspecives on Music Cognition*. The AAAI Press and The MIT Press, 1992. ISBN 0-262-52170-9.

[DHdR] Peter Desain, Henkjan Honing, and Klaus de Rijk. The quantization of musical time: a connectionist approach. In *Music, Mind and Machine: Studies in Computer Music, Music Cognition and Artificial Intelligence*, pages 61–78.

[Dow78] W. Jay Dowling. Scale and contour: Two components of a theory of memory for melodies. *Psychological Review*, 85(4):341–354, July 1978.

[GLCS95] Asif Ghisa, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by humming: Musical information retrieval in an audio database. In *Proceedings of ACM Multimedia*, pages 231–236. The Association for Computing Machinery, 1995.

[Jac91] Arthur Jacobs, editor. *The Penguin Dictionary of Music*. Penguin Books, 5 edition, 1991. ISBN 0-14-051159-8.

[KBWW95] Douglas Keislar, Thom Blum, James Wheaton, and Erling Wold. Audio analysis for content-based retrieval. In *Proceedings of International Computer Music Conference*, pages 199–202, 1995.

[KMT93] Tetsuya Kageyama, Kazuhiro Mochizuki, and Yosuke Takashima. Melody retrieval with humming. In *Proceedings of International Computer Music Conference*, pages 349–351, 1993.

[LA96] Joo Ho Lee and Jeong Soo Ahn. Using n-grams for korean text retrieval. In *SIGIR '96, Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 216–224. The Association for Computing Machinery, 1996.

[Lin77] Harry B. Lincoln. Encoding, decoding and storing melodies for a data base of renaissance polyphony: A progress report. In *Proceedings of the third international conference on Very Large Data Bases*, pages 277–282, October 1977.

[McC76] Edward M. McCreight. A space-economical suffix tree construction algorithm. *Journal of the Association of Machinery*, 23(2):262–272, April 1976.

[MID] MIDI Manufacturers Association, MMA, PO Box 3173, La Habra, CA 90632-3173. *The Complete MIDI 1.0 Detailed Specification*.

[MR93] Bernard Mont-Reynaud. Seemusic: A tool for music visualization. In *Proceedings of International Computer Music Conference*, pages 457–460, 1993.

[MS90] Marcel Mongeau and David Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, June 1990.

[MSW+96] Rodger J. McNab, Lloyd A. Smith, Ian H. Witten, Clare L. Henderson, and Sally Jo Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the 1st ACM international conference on Digital libraries*, pages 11–18. The Association for Computing Machinery, 1996.

[NTJ+96] Shiho Nobesawa, Junya Tsutsumi, Sun Da Jiang, Tomohisa Sano, Kengo Sato, and Masakazu Nakanishi. Segmenting sentences into linky strings using d-bigram statistics. In *COLING-96: Proceedings of the 16th International Conference on Computational Linguistics*, volume 2, pages 586–591, 5–9August 1996.

[Rad96] Gary M. Rader. Creating printed music automatically. *IEEE Computer*, 29(6):61–68, June 1996.

[Sch91] Manfred Schroeder. *Fractals, Chaos, Power Laws: minutes from an infinite paradise.* W. H. Freeman and Company, 1991. ISBN 0-7167-2136-8.

[Sch92] Helmut Schaffrath. The retrieval of monophonic melodies and their variants. In *Computer Representation and Models in Music*, pages 95–109. Academic Press Ltd., 1992. ISBN 0-12-473545-2.

[SF97] Eleanor Selfridge-Field. Introduction: Describing musical information. In Eleanor Selfridge-Field, editor, *Beyond MIDI: The Handbook of Musical Codes*, chapter 1, pages 3–38. The MIT Press, 1997. ISBN 0-262-19394-9.

[Sht94] S. Shtrikman. Some comments on Zipf's law for the Chinese language. *Journal of Information Science*, 20(2):142–143, 1994.

[SM83] Gerald Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983. ISBN 0-07-Y66256-5.

[Smi96] Terence R. Smith. A digital library for geographically referenced materials. *IEEE Computer*, 29(5):54–60, May 1996.

[SS68] Herbert A. Simon and Richard K. Sumner. Pattern in music. In Benjamin Kleinmuntz, editor, *Formal Representation of Human Judgement*, chapter 8, pages 219–250. 1968.

[Sue86] Ching Y. Suen. *Computational studies of the most frequent Chinese words.* World Scientific, 1986. ISBN 9971-50-022-1.

[SW97] Sean M. Smith and Glen N. Williams. A visualization of music. In *Proceedings of the conference on Visualization '97*, pages 499–503, 1997.

[Tay92] Eric Taylor. *The AB Guide to Music Theory, Part II.* The Associated Board of the Royal Schools of Music (Publishing) Limited, 1992. ISBN 1-85472-447-9.

[Tay93] Eric Taylor. *The AB Guide to Music Theory, Part I.* The Associated Board of the Royal Schools of Music (Publishing) Limited, 1993. ISBN 1-85472-446-0.

[The81] *The Ultimate Fake Book.* Hal Leonard Publishing Corporation, "C" instruments edition, 1981. ISBN 0-9607350-0-3.

[WBKW96] Erling Wold, Thom Blum, Douglas Keislar, and James Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, Fall 1996.

[Wil96] Robert Wilensky. Toward work-centered digital information services. *IEEE Computer*, 29(5):37–44, May 1996.

[WKSS96] Howard D. Wactlar, Takeo Kanade, Michael A. Smith, and Scott M. Stevens. Intelligent access to digital video: Informedia project. *IEEE Computer*, 29(5):46–52, May 1996.

[YK98a] Chi Lap Yip and Ben Kao. Finding index terms from text documents. In *First Asia Digital Library Workshop — East meets West (ADL 1998).* The University of Hong Kong Libraries, The University of Hong Kong Libraries, August 1998. ISBN 962-85355-1-X.

[YK98b] Chi Lap Yip and Ben Kao. Indexing multilingual documents on the web. In *COMPSAC 1998, Proceedings of the twenty-second annual international Computer Software and Applications Conference*, pages 576–581. IEEE Computer Society, August 1998. ISBN 0-8186-8585-9; ISSN 0730-3157.

[Zip65] George Kingsley Zipf. *The Psycho-Biology of Langauge*. The MIT Press, first MIT paperback edition, August 1965.