Harmonic Models for Polyphonic Music Retrieval

Jeremy Pickens
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst
jeremy@cs.umass.edu

Tim Crawford
Music Department
Kings College London
tim.crawford@kcl.ac.uk

ABSTRACT

Most work in the ad hoc music retrieval field has focused on the retrieval of monophonic documents using monophonic queries. Polyphony adds considerably more complexity. We present a method by which polyphonic music documents may be retrieved by polyphonic music queries. A new harmonic description technique is given, wherein the information from all chords, rather than the most significant chord, is used. This description is then combined in a new and unique way with Markov statistical methods to create models of both documents and queries. Document models are compared to query models and then ranked by score. Though test collections for music are currently scarce, we give the first known recall-precision graphs for polyphonic music retrieval, and results are favorable.[1]

1. INTRODUCTION

Music retrieval is a rapidly growing field. While traditionally not part of the information retrieval task, music retrieval is receiving increased attention. The availability of online collections and numerous digital library projects for music has fueled a need for effective searches. However, the vast majority of work in music information retrieval has focused on the the monophonic domain. Most of the interesting, as well as widely available, music is polyphonic. Polyphony, with its multidimensional sequences of overlapping notes, is much more complex (see Figure 1). Techniques used for text and monophonic music, both of which can be thought of as one-dimensional sequences of features, cannot be directly applied to polyphonic music. These techniques include string-matching and straightforward (n-gram) Markov modeling.

We therefore present a method for polyphonic music retrieval which begins by preprocessing a music score to analyze its underlying harmonic structure probabilistically, based on key-distance measures derived from Carol Krumhansl's work on inter-key relationships [8]. The output of this analysis is a probability distribution over all chords, one distribution for each *simultaneity* occurring in the score. The use of all chords to characterize a polyphonic music sequence, rather than a single chord reduction, is a distinctive trait of our system.

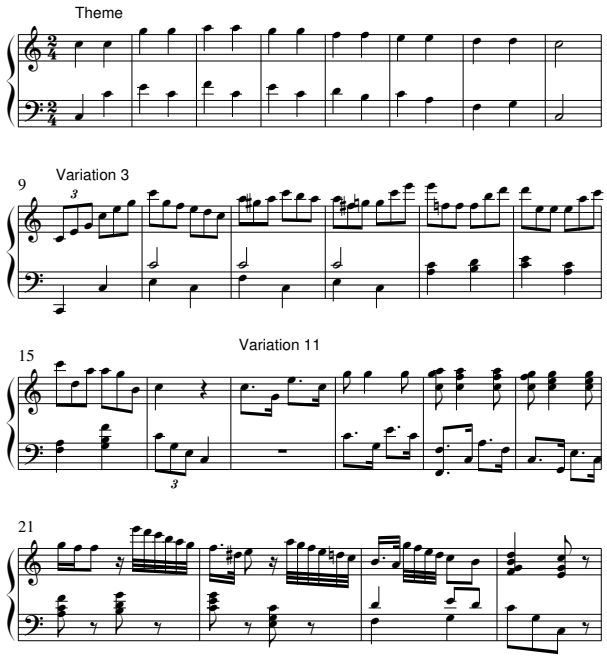

Figure 1: Sample variation excerpts, from the Mozart 'Twinkle, twinkle, little star' composition

Once the harmonic description is complete, simple Markov modeling techniques using various fixed-sized Markov chain lengths are applied. The manner in which the harmonic description is combined with Markov statistical techniques is also novel. The document is then indexed as this fixed-size Markov model. This is done for every document in the collection, prior to retrieval. Queries are then modeled using the same modeling technique as the documents. Finally, a scoring function is used to calculate the dissimilarity between a query model and each document model in the collection, and the documents are ranked by this score.

It should be stressed that our methods do not seek to produce a formal music-theoretical harmonic analysis of a score,

[1]This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-9905842. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

but to output a pattern of harmonic probabilities which are hopefully characteristic of variations to which a score is judged relevant. We justify our methods emperically; the recall-precision results in Section 5.2, the first of their kind for polyphonic music retrieval, are good.

2. BACKGROUND

Terminology We begin by defining a number of terms used in this paper. The first is the domain itself in which we are working: music. Within this paper, all music with which we work comes from the symbolic, rather than audio, domain. This means that the exact onset time, pitch, and duration of every single note in a piece of music are known. This amount of structure is not explicitly available from an audio source, such as a WAV or MP3 file, but is found in formats such as MIDI (www.midi.org) and Kern (www.musedata.org).

Within the music domain, we must define *monophony*, *homophony*, and *polyphony*. Monophonic music has at most one note playing at any given time; before a new note starts the previous note must have ended. Homophonic music has at most one *set* of notes playing at any given time. For any set of notes that start at the same time, no new note or notes may begin until every note in that set has ended. Polyphonic music has no such restrictions. Any note or set of notes may begin before any previous note or set of notes has ended, which proves difficult for any clear, unambiguous sense of sequentiality.

The next term we define is *simultaneity*. A simultaneity is an octave-invariant (mod 12) pitch set. We use this name because simultaneities are created from polyphonic music by extracting *at a given point in time* either all notes which *start* at that point in time [4], or all notes which are *sounding* at that point in time [10]. Though these two techniques are the most common, other researchers have used larger time or rhythm-based windows from which to extract simultaneities (see Pickens [14] for further discussion).

We define a *lexical chord* as a codified pitch template. Of the 12 octave-equivalent (mod 12) pitches in the Western canon, we select some n-sized subset of those, call the subset a *chord*, give that chord a name, and add it to the lexicon. Not all possible chords belong in a lexicon; with $\binom{12}{n}$ possible lexical chords of size n, and 12 different choices for n, we must restrict ourselves to a musically-sensible subset.

Finally, we define *harmonic description* as the process of fitting simultaneities to lexical chords. A number of researchers have focused exclusively on the harmonic description task. Prather [16] selects the most salient lexical chord from a simultaneity by examining neighboring simultaneities to eliminate ambiguity. Chou [2] uses clues such as frequency and consonancy to select the most salient lexical chord. Pardo [12] uses notions of harmonic similarity or consistency to dynamically shape the size of the simultaneities, so that partitioned areas are created in positions where a single (harmonically significant) lexical chord dominates.

While not a comprehensive list of harmonic description papers, these are indicative of the effort we see to create descriptions of music in which only the most salient lexical chord is used. The difference in our technique is that all chords describe the music, to varying degrees; the purpose of our harmonic description is to determine to what extent each chord fits. But no chord is eliminated completely, no matter how unlikely. So while we are not the first in the music IR community to suggest harmonic description, we are the first that we know of to create harmonic *distributions* (using all available chords to characterize a set of notes) rather than harmonic *reductions* (using a single chord to characterize a set of notes), and apply those distributions to the ad hoc retrieval task.

The Language Modeling Approach Language Modeling (LM) has received much attention recently in the text information retrieval community. It is only natural that we wish to leverage some of the advantages of LM and apply it to music.

> "The approach to retrieval taken here is to infer a language model for each document and to estimate the probability of generating a query according to each model. The documents are then ranked according to these probabilities...The advantage of using language models is that observable information, i.e., the collection statistics, can be used in a principled way to estimate these models and do not have to be used in a heuristic fashion to estimate the probability of a process that nobody fully understands...When the task is stated this way, the view of retrieval is that a model can capture the statistical regularities of text without inferring anything about the semantic content." Ponte [15]

Even though our retrieval task is polyphonic music rather than text, we are duplicating the LM framework by creating statistical models of each piece of music in a collection and then ranking the pieces by those statistical properties. Thus, while it might be more appropriate to name this work "statistical music modeling", we still say that we are taking the language modeling *approach* to information retrieval. We attempt to "capture the statistical regularities of [music] without inferring anything about the semantic content".

To our knowledge, the first LM approach to music IR was done in the monophonic domain [13]. Other techniques, which we also roughly categorize as taking the LM approach, apply 1^{st}-order Markov modeling to monophonic note sequences [17, 7]. Another technique extends the modeling to the polyphonic domain, using both 0^{th} and 1^{st}-order Markov models of raw note simultaneities to represent scores [1].

3. HARMONIC DESCRIPTION

3.1 Step 1: Simultaneity creation

The first step in this process is to reduce complex polyphonic music to a sequence of simultaneities. We do this by ignoring durational information and adding to each simultaneity all pitches of notes which start at the same time.

3.2 Step 2: Selection of Chord Lexicon

The chord lexicon used in this paper is the set of 24 major and minor triads, one each for all 12 members of the chromatic scale: C Major, c minor, C♯ Major, c♯ minor ... B♭ Major, b♭ minor, B Major, b minor. No distinction is made between enharmonic equivalents (C♯/D♭, A♯/B♭, E♯/F, and so on). Assuming octave-invariance, the three members of a major triad have the relative semitone values n, $n+4$ and $n+7$; those of a minor triad n, $n+3$ and $n+7$.

The 24 lexical chords can be viewed as points on two overlapping 'circles of fifths', one for major triads, the other for

minor triads. Each circle is constructed by placing chords adjacently whose root pitch is separated by the interval of a fifth (7 semitones); for example, G Major or minor (root pitch-class 7) has immediate neighbours C (7 - 7 = 0) and D (7 + 7 = 14, i.e. octave-invariant pitch-class 2). Thus each Major tonic chord (G major, say) stands in appropriately close proximity to its dominant (D Major) and subdominant (C Major) chords, i.e. those to which it is most closely related in music-theoretical terms. The two circles (Major and minor) may be aligned by placing Major triads close to their respective relative minor triads, as shown in Figure 2 (Major triads are shown in upper case, minor triads in lower case).

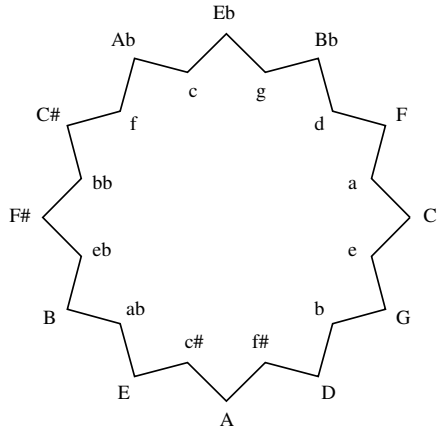

Figure 2: Lexical chords and their relative distances

While it is clear that the harmony of all but the crudest music cannot be reduced to a mere succession of major and minor triads, as this choice of lexicon might be thought to assume, we believe that this is a sound basis for a probabilistic approach to harmonic description. As our goal is not the selection of a single, most salient lexical chord, but a distribution over possible harmonic chords, we feel that triads are large enough to distinguish between harmonic patterns, but small enough to robustly accomodate harmonic invariance. Furthermore, our approach depends on a reliable and easily applied measure of 'distance' between lexicon members which is satisfactory on the grounds of music theory and corresponds with practical experience. Such a measure indeed exists for these 24 major and minor triads, but not for more complex chords.

During the 1970s and 1980s the music-psychologist Carol Krumhansl conducted a ground-breaking series of experiments into the perception and cognition of musical pitch [8]. By using the statistical technique of multi-dimensional scaling on the results of experiments on listeners' judgements of inter-key relationships, she produced a table of coordinates in four-dimensional space which provides the basis for the lexical chord distance measure we adopt here. The 'distance' between triads a and b can be expressed as the four-dimensional Euclidean distance between these coordinates, where w_x, x_x, y_x and z_x are the four coordinates of triad x as given in Krumhansl's table (which we do not reproduce here):

$$EDist(a,b) = \sqrt{\begin{array}{ccccc}(w_a - w_b)^2 & + & (x_a - x_b)^2 & + \\ (y_a - y_b)^2 & + & (z_a - z_b)^2 & \end{array}} \quad (1)$$

As Krumhansl demonstrates, her results, derived from controlled listening tests, correspond very closely with music theory.

3.3 Step 3: Harmonic Description

Once again, harmonic description is the process of fitting simultaneities to lexical chords. Given a simultaneity s, we count, for each chord c in the lexicon, the number of pitches in common between the simultaneity and the lexical chord: $Num(s,c)$. Normalizing by the total overlapping pitch count would give us a naive distribution for s over the lexicon. While retrieval still may be possible using this distribution, it fails to take into account harmonic similarities between nearby lexical chords. We therefore need to smooth these initial probability estimates.

Harmony, as musicians perceive it, is a highly contextual phenomenon which depends on many factors, only one of which is the simple pitch count we have performed thus far. It is common for non-harmonic notes (that is, notes that do not 'belong to' the prevailing harmony) to occur in a simultaneity. We need a technique for lowering the probability mass of those lexical chords in which non-harmonic notes occur, while raising the probability mass of lexical chords in which the 'truer' harmonies do occur.

It is necessary, therefore, for us to account for the presence of all pitches in simultaneity s in terms of the harmonic context of s. Since, in general, 'contributions' of near neighbours in terms of inter-key distance are preferred, we use that fact as the basis for computing a suitable context. We effectively smooth the distribution by summing contributions to the harmonic description from all members of the lexicon, even those that are not explicitly represented in the current simultaneity. For each lexical chord c its contribution will be proportional to the number of pitches shared with the simultaneity s and inversely proportional to its inter-key distance from the lexical chord p, whose context is being summed. This is given by the following equation:

$$Context(s,c) = \sum_{p=1}^{24} \frac{Num(s,p)}{Edist(p,c)+1} \quad (2)$$

This context score is computed for every chord in the lexicon (each point in the distribution), and then the entire distribution is normalized by the sum total. In addition to providing a better estimate for the harmonic distribution, this 'smoothed' context also accounts for more complex input chords, such as 7^{th} chords, by retaining higher probability mass for those nearby triads in which a 7^{th} chord 'participates'.

Currently, this harmonic smoothing is performed within the scope of a single simultaneity, not over time. In future experiments we shall extend the harmonic context to take into account the profile of neighboring simultaneities. We propose in the future to adopt a time-window-based approach, summing the lexical contributions in the way described above across events within the window in inverse proportion to their time-, event-, or beat-based distance from the current simultaneity, with additional weightings provided according to metrical stress, note duration or other factors that might be considered helpful. Indeed, harmonic smoothing, properly executed, might be a way of integrating the problematic, not-quite-orthogonal dimensions of pitch and duration within a polyphonic source. A larger, time-based smoothing might also yield a richer harmonic descrip-

tion, because it gives less weight to transient changes in harmony arising from non-harmonic notes such as passing tones or appoggiaturas.

4. MARKOV MODELING

Markov models are often used to capture statistical properties of a state sequence over time. We want to be able to predict future occurrences of a state by the presence of sequences of previous states.

4.1 Model Estimation

For our harmonic approach, we have chosen lexical chords as the states of the Markov model. For an n^{th}-order model, a $24^n \times 24$ matrix is constructed, with the rows representing the *previous state* space, and the columns representing the *current state* space. An $(n+1)$ sized window slides over the sequence of lexical chord distributions and Markov chains are extracted from that window. The count of each chain is added to the matrix, where the crossproduct of the first n states is the previous state, and the $(n+1)^{th}$ state is the current state. Finally, when the entire observable sequence has been counted, each row of the matrix is individually summed and the elements of each row normalized by the sum total for that row.

With Markov model estimation over typical data, input sequences are one-dimensional, so there is never more than one Markov chain, or path, per window. Our approach is slightly different. Recall from Section 3 that instead of a one-dimensional sequence of states, we have a sequence of 24-point distributions. So our solution is to assume independence between points in each distribution at each timestep, so that an exhaustive number of independent, one- dimensional paths through the distribution sequence may be traced. (This exhaustive branching paths approach is abstractly similar to one suggested by Doraisamy [3]. The context is somewhat different; we use lexical chord distributions rather than notes, but the basic idea is similar.)

Within a given window, there are 24 different ways (24 lexical chords) to select the first element in the path. There are also 24 possibilities for the second element, and so on, up to the size of the window. A simple recursive algorithm, which we consider trivial enough not to duplicate here, assembles all such paths. Each path, thus constructed, is not counted as one full observation. Instead, observations are proportional; the degree to which each path is observed is a function of the amount by which all elements of the path are present. Since independence between neighboring simultaneities was assumed, this becomes the product of the values of each state which comprises the path.

4.2 Example

The figure below is an example of the type of output of our harmonic description. In the interest of space, we assume a lexicon of only three chords, to which we arbitrarily assign the names P, Q, and R. These lexical chords are arranged circularly, with Q following P, R following Q, and P following R. We use this harmonic description to create a 1^{st}-order Markov model, which means that the probability of being in the current state is dependent only on the previous state.

We first obtain an accurate count of the Markov chains over the entire sequence. We begin the window from timestep 1 to timestep 2. The sequence $P \rightarrow P$ is observed in proportion to the amount in which one is *in* P at timestep 1 and also *in* P at timestep 2 (0.2 * 0.1 = 0.02). The sequence $Q \rightarrow R$ is observed in proportion to the amount in which one is in P at timestep 1 and then in Q at timestep 2 (0.5 * 0.8 = 0.4), and so on. The entire timestep 1 to timestep 2 window is illustrated in Figure 3.

Lexical Chord	Timestep (Simultaneity) 1	2	3	4	5
P	0.2	0.1	0.7	0.5	0
Q	0.5	0.1	0.1	0.5	0.1
R	0.3	0.8	0.2	0	0.9

$P \rightarrow P$	=	0.2 * 0.1	=	0.02
$P \rightarrow Q$	=	0.2 * 0.1	=	0.02
$P \rightarrow R$	=	0.2 * 0.8	=	0.16
$Q \rightarrow P$	=	0.5 * 0.1	=	0.05
$Q \rightarrow Q$	=	0.5 * 0.1	=	0.05
$Q \rightarrow R$	=	0.5 * 0.8	=	0.4
$R \rightarrow P$	=	0.3 * 0.1	=	0.03
$R \rightarrow Q$	=	0.3 * 0.1	=	0.03
$R \rightarrow R$	=	0.3 * 0.8	=	0.24
		TOTAL	=	1.0

Figure 3: Example full set of 1^{st}-order (bigram) transitions from timestep 1 to timestep 2

It should be emphasized that these values are not the probabilities of the final model at this stage; they are counts. That the counts are not integers does not matter. Think of it this way: Suppose 100 musicians were each given an instrument which was only capable of playing one lexical chord at a time (such as an autoharp), and not arbitrary individual notes. Suppose further that these musicans were given sheet music of the documents which we are modeling, which documents contain only individual notes and not chords. The musicians would be forced to make a choice about which lexical chord to play. We are saying that two of the musicians would then choose to play $P \rightarrow P$, forty would play $Q \rightarrow R$, and so on. Our actual observation of what lexical chords were played includes every single one of these possibilities, concurrently and independently. It is as if this one snippet of music were played 100 times, 9 different ways, and we simply count the number of times each way was played. Divide those integer counts by 100 and the proportion, and thus the final model, remains the same. (This differs from the *hidden* Markov model approach in that we do not assume there is a single chord or other hidden state which is the one "real" way to play the snippet.)

Next, we add these transition counts to our count matrix, and repeat the process for timesteps $2 \rightarrow 3$, $3 \rightarrow 4$, and $4 \rightarrow 5$ (the remainder of the piece). When finished, we follow the standard method of transforming the matrix into a Markov model by normalizing the outgoing arcs from each state, so that the total score sums to 1.0, as shown in Figure 4.

Count Matrix			
	P	Q	R
P	0.44	0.43	0.63
Q	0.17	0.16	0.87
R	0.69	0.21	0.40

Markov Model			
	P	Q	R
P	0.293	0.287	0.42
Q	0.1417	0.1333	0.725
R	0.5308	0.1615	0.3077

Figure 4: Example 1^{st}-order count matrix (*left*) and normalized Markov model (*right*)

4.3 Transposition Invariant Model Estimation

In Section 1 we spoke about sources of variation in music, and how variation built around relatively stable harmonic patterns is common. Another important source of variation is key transposition, in which an entire piece of music is shifted up or down by a number of semitones. This has major consequences in our harmonic description in that if the key is shifted, the patterns of lexical chords found at each timestep may vastly differ. This has the potential to break our retrieval system.

The favored solution to key transposition in monophonic music is to use the semitone interval between contiguous notes, rather than the absolute pitches of the notes themselves. Our solution is to use the *spread*, in steps around the circle from Figure 2, between lexical chords at contiguous timesteps. We count as equivalent all those neighboring (timestep adjacent) chords which have the same spread. Absolute chord paths of length 2, 3 and 4 become relative *chord-spread* paths of length 1, 2 and 3.

Returning to our example from Figure 4, the spread from $P \rightarrow Q$ (+1 or -2) is the same as the spread from $Q \rightarrow R$ (+1 or -2), so these two chord pairs are equivalent. Since the lexical chords are arranged in a circle, every positive value has an equivalent negative value. Going up by one is the same as going down by the size of the lexical circle minus one. We therefore create Markov matrices in terms of the positive spreads only, for simplicity. Figure 5 is the example from Figure 4 recast as a transposition invariant model.

Spread	Equivalent Transitions	Transn. Counts	Count Totals	Markov Model
+0	$P \rightarrow P$	0.44		
	$Q \rightarrow Q$	+ 0.16	= 1.00	0.25
	$R \rightarrow R$	+ 0.40		
+1	$P \rightarrow Q$	0.43		
	$Q \rightarrow R$	+ 0.87	= 1.99	0.4975
	$R \rightarrow P$	+ 0.69		
+2	$P \rightarrow R$	0.63		
	$Q \rightarrow P$	+ 0.17	= 1.01	0.2525
	$R \rightarrow Q$	+ 0.21		

Figure 5: Example 0^{th}-order transposition invariant count matrix and Markov model

4.4 Scoring Function

At index time, a model of every piece of music in the collection is created and stored. At query time, a query model is created with exactly the same structure as the collection models. If the collection was modeled using 2^{nd}-order transposition invariant Markov models, the query must be modeled in the same fashion.

Our goal is to produce a ranked list for a query across the collection. We wish to rank those pieces of music at the top which are most similar to the query, and those pieces at the bottom which are least similar. In order to do this we need a scoring function. We have chosen as this function the Kullback-Liebler (KL) divergence, a measure of how different two distributions are, over the same event space. The divergence is always zero if two distributions are exactly the same, or a positive value if the distributions differ. We denote the KL divergence between query model q and music document model d as $D(q||d)$. "The KL divergence between $[q]$ and $[d]$ is the average number of bits that are wasted by encoding events from a distribution $[q]$ with a code based on the not-quite-right distribution $[d]$" [11].

In a Markov model, each previous state, each row in the $24^n \times 24$ matrix, is a complete distribution. We therefore compute a divergence score for each row in the model, and add the value to the total divergence score for that query-document pair. This is given by the following equation, where q_i and d_i represent each previous state.

$$D(q||d) = \sum_{q_i \in q, d_i \in d} \left(\sum_{x \in X} q_i(x) \log \frac{q_i(x)}{d_i(x)} \right) \quad (3)$$

4.5 General Music "Back off" Model

There is a problem in that sometimes a document model can have estimates of zero probability. This is due to limitations of finite computers. Floating point or double values can only hold so much information, and probability values can be so small as to become zero. The divergence score in such cases ($q_i(x) \log \frac{q_i(x)}{0}$)) automatically goes to infinity. This small problem in just a single value could therefore throw off our entire score for that document. We therefore must create some small but principled non-zero value for every document model zero value. There are many ways to do this, but we have done so by "backing off" to a general music model, using the value of that previous state node from the general model whenever we encounter a zero value in any particular document model.

A general music model is created by averaging the models over the entire set of document models in the collection. In principle, there could still remain zero values in the general music model, depending on the size and properties of the collection. In our experiments, however, we found this almost never to be the case. Also, it should be observed that when the query model has a zero probability in any cell, there is no problem. The KL divergence for that point is $0 \log \frac{0}{d_i(x)}$, which is zero.

5. EVALUATION

5.1 Source Collection and Query Sets

The basic test collection on which we tested our retrieval method was assembled from data provided by the Center for Computer Assisted Research in the Humanities, Stanford University. It comprises around 3000 files of separate movements from polyphonic fully-encoded music scores by a number of classical composers, including a significant proportion of the works of J.S. Bach, of varying keys, textures (i.e. average numbers of notes in a simultaneity) and lengths (numbers of simultaneities). To this basic collection we add, for the purposes of the present paper, three additional sets of polyphonic music data:

(1) 26 individual variations on the tune known to English speakers as 'Twinkle, twinkle, little star' (in fact a mixture of mostly polyphonic and a few monophonic versions); (2) 22 versions of John Dowland's 'Lachrimae Pavan', collected as part of the ECOLM project [9] from different 16th and 17th-century sources, sometimes varying in quality (numbers of 'wrong' notes, omissions and other inaccuracies), in scoring (for solo lute, keyboard or five-part instrumental ensemble), in sectional structure and in key; (3) 17 individual variations on the well-known baroque tune 'Les Folies d'Espagne' from a little-known English lute manuscript now held in the Poznan University Library, Poland [6], another ECOLM encoding.

	1. Twinkle	2. Lachrimae	3. Folia
Number of variations	26	22	17
Key profiles	C(17), A(6), c(1), Unknown(2)	g(15), d(2), a(5)	d(17)
Note consistency	Low	Low	Medium
Harmonic-profile consistency	Medium	Medium	High

Table 1: Query descriptions

In each case, the set of score-files shares a relatively low note consistency (the number of and actual notes played), which makes string-matching or n-gramming approaches even more difficult and our harmonic modeling necessary. Nevertheless, each set of score-files also shares a broadly similar harmonic profile, although there are some sophisticated variations by Mozart in set (1) which deviate somewhat from the 'norm'. Some of the files in (2) present different settings whose detailed harmonic progressions are different from the rest, although the broader overall profile remains similar.

For our experiments, we use standard Cranfield-style evaluation, with queries and relevance judgements across the collection, assembled as follows: For a given set of variations (Twinkle for example), a single variation is selected as the user query. The remaining n variations (25 in this case) are tagged as relevant to this query, and inserted into the collection. The remainder of the collection is assumed to be non-relevant. A ranked list is created for this query. The query is then re-inserted into the collection as a 'relevant' document, and a different variation removed and selected as the new query. This is repeated for all 26 variations, each with a slightly different set of 25 relevant documents. (It is quite feasible that a user could present any one of the documents as a query, and expect to retrieve all 25 remaining variations.) Depending on which variation is used as the query, however, the resulting ranked list can vary wildly. The average across all 26 ranked lists is taken; these are the recall-precision graphs we present in the next section.

5.2 Results

Interpolated 11-point recall-precision graphs for all the queries are found in Figures 6 through 8. These graphs show both regular harmonic-based 0^{th} to 3^{rd}-order Markov models (MM0-MM3) and transposition invariant harmonic-based 0^{th} to 2^{nd}-order Markov models (TIMM0-TIMM2). Average precision (non-interpolated) for these same queries is given in Table 2.

These results show a number of different patterns. There is significant improvement over the baseline (random ranking) in every case but the transposition invariant Folia queries. As a concrete example, consider the variations in Figure 1 at the beginning of this paper. In the case where the Theme is used as a query (under a harmonic MM0 model) Variation 11 was ranked sixth and Variation 3 was ranked twelvth. When Variation 3 was used as the query, variation 11 was ranked sixth and the Theme was ranked eighth. Again, this is from a collection of 3000 similar-genre, similar era pieces. Though the pieces contain significant variation, our harmonic modeling captures the similarity.

We next observe the tradeoff between regular and transposition invariant modeling. Music IR literature from the monophonic domain shows that as basic lexical units become more generalized, precision suffers [5]. Thus it is only natural to expect that transposition invariance opens up the possibility that not only will we get more relevant documents, but we will get more non-relevant documents as well.

This pattern is present in all three query sets. As we see in Table 1, the Twinkle and Lachrimae queries had a good number of key-transposed variations, so when compared to the MM results, the precision of the TIMM results drops at low recall, but increases at higher recall. The Folia queries had no key-transposed variations (see Table 1), so transposition invariant modeling only dropped precision without boosting recall.

Furthermore, it is interesting to realize that, while the Folia query set had the least amount of key-transposed variations, the Lachrimae set had the most. Our results match this fact; the transposition invariant Lachrimae queries have the least drop in precision and the highest gain in recall of all three query sets. The Twinkle queries were somewhere in the middle, again as expected.

Finally, some discussion about the different fixed-size models is necessary. Here, however, the trends are not as clear. We offer the following observation, but leave the readers with graphs to decide for themselves. For the regular Markov models, the longer the model, the better the results, with the exception of the 0^{th}-order models which generally outperform them all. Our harmonic description by itself apparently goes a long way in helping distinguish relevant and non-relevant documents. For the transposition invariant models, on average, the longer models do slightly better than the shorter models. This would make sense, as much of the existing music IR literature suggests that the more generalized the lexical unit, the longer the sequence needs to be in order to distinguish relevant from non-relevant documents.

6. CONCLUSION

There are many kinds of variation in music, and many ways of accommodating that variation. Pitch intervals capture key-transposition invariance for monophonic music. Duration intervals capture rhythmic invariance for monophonic music. Our modeling-based retrieval system was an attempt to capture harmonic invariance for polyphonic music. We started with the realistic assumption that in many real-world situations, variations on a piece of music may include many different pitches but the underlying harmonies remain similar.

From this assumption came the problem of lexical chord selection. Major and minor triads were chosen for their robustness in describing but not overspecifying harmony. Lexical chords which are too large (every known 11th-chord, for example) are too specific. One might as well use the entire raw simultaneity; both are not tolerant of missing or extra notes, which we consider a fatal flaw. We also did not want lexical chords which were too small (dyads, for example). Under such general chords the harmonies would blend together undistinguishably. Our choice of major and minor triads was meant to be robust enough to handle harmonic variation.

Yet even with triads there is not a perfect fit to every simultaneity. Therefore, we made the decision to do harmonic description not as a selection of a single, most salient

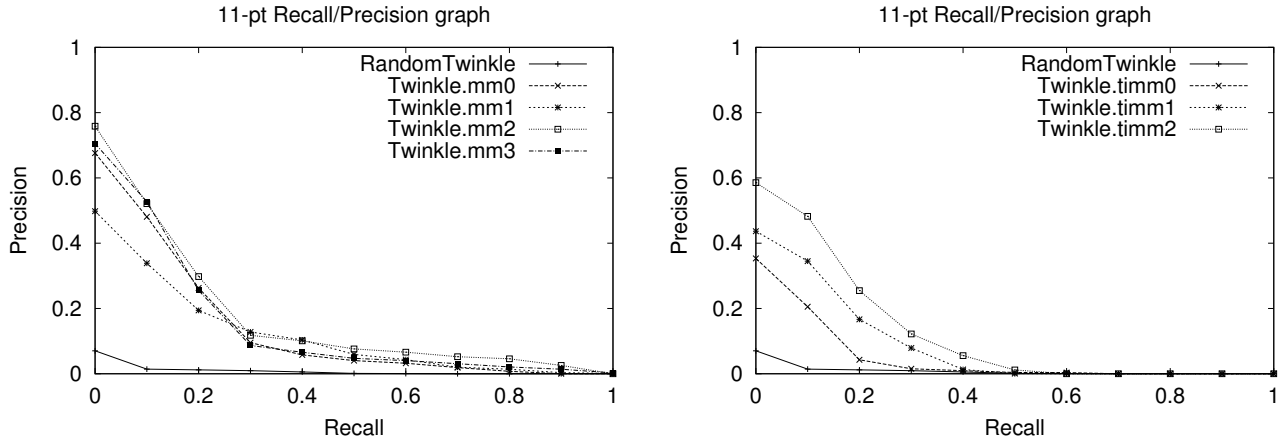

Figure 6: **Twinkle queries: Markov model (*left*), transposition invariant Markov model (*right*)**

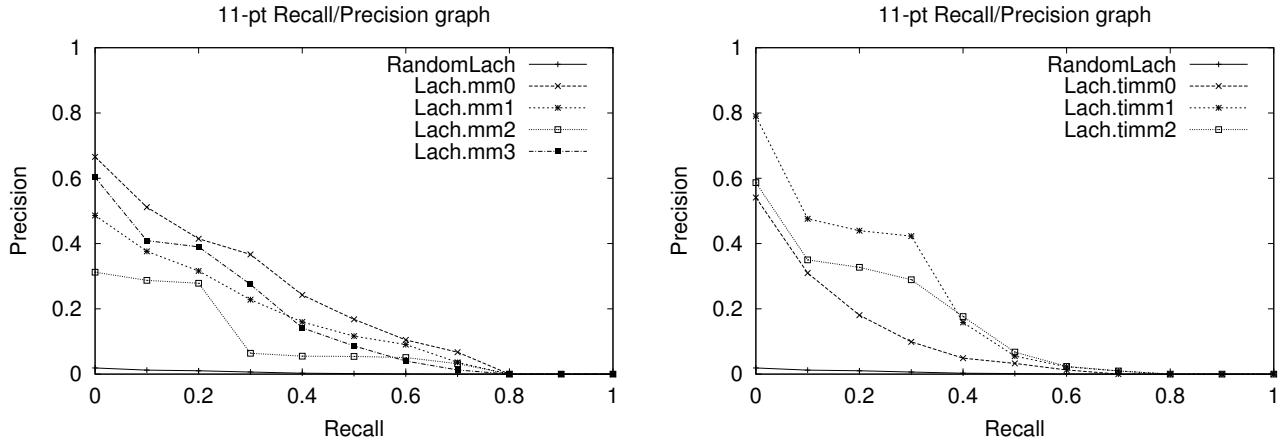

Figure 7: **Lachrimae queries: Markov model (*left*), transposition invariant Markov model (*right*)**

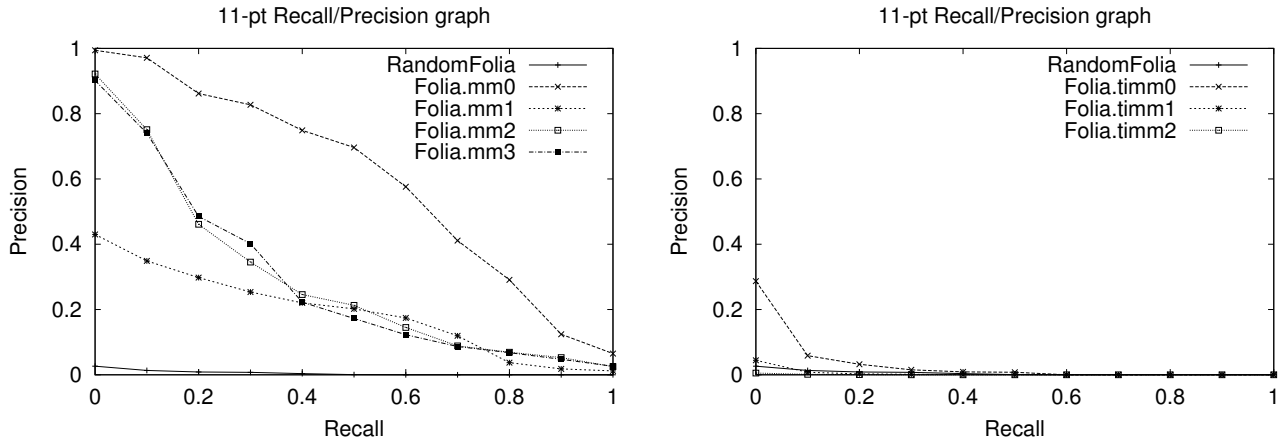

Figure 8: **Folia queries: Markov model (*left*), transposition invariant Markov model (*right*)**

	Random	mm0	mm1	mm2	mm3	timm0	timm1	timm2
Twinkle Queries	0.0093	0.1205	0.1094	0.1517	0.1332	0.0414	0.0772	0.1103
Lachrimae Queries	0.0039	0.2144	0.1429	0.0883	0.1649	0.0967	0.1961	0.1461
Folia Queries	0.0045	0.5904	0.1669	0.2801	0.2759	0.0286	0.0038	0.0005

Table 2: Average precision (non-interpolated) over all relevant documents

lexical chord, but as a distribution of chords. As far as we know, this choice is new in the field of music information retrieval. Using a distribution in this manner also fits with the Language Modeling approach by letting the statistical regularities of the source document speak for themselves, rather than trying to semantically infer that a composer truly meant for a single chord to represent the piece at any particular point.

Many of the general topic areas touched on by our system are available elsewhere, but the individual details and, in particular, the combination of these areas, is unique to this work. Simultaneity creation, lexical chord selection, harmonic description, Markov modeling, exhaustive combinatorial path extraction (to a fixed length), and transposition invariance all combine to create a working, probabilistic, fully-polyphonic music retrieval system.

7. FUTURE WORK

The two parts of this retrieval approach which will be explored and refined in the future are the creation of simultaneities and the fitting (and smoothing) of those simultaneities to the lexical chord set. These two tasks are not entirely independent; how the simultaneities are chosen has a large influence on the smoothing techniques used for lexical chord fits. Improvements in these areas will, we believe, improve the recall-precision results obtained with our system.

Furthermore, early experiments indicate that our harmonic modeling is robust when applied to imperfect transcriptions of polyphonic audio. A harmonic model of an imperfect audio transcription still allows us to retrieve the original piece at an extremely high rank. This bodes well for the future integration of polyphonic audio and symbolic music, using one to search the other and vice versa. This is an important step in today's online digital world.

8. ACKNOWLEDGEMENTS

We would like to thank Dawn Lawrie and Victor Lavrenko for discussion of evaluation methodology and Markov modeling, respectively. We are especially grateful to Eleanor Selfridge-Field, Craig Sapp, Bret Aarden, and David Huron for their assistance with the source collection and Kern music data format. Finally, we thank Don Byrd and Matthew Dovey for continued discussion on topics relevant to this paper.

9. REFERENCES

[1] W. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. Bartsch, D. Bykowski, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart: Music retrieval via aural queries. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 73–81, Indiana University, Bloomington, Indiana, October 2001.

[2] T. C. Chou, A. L. P. Chen, and C. C. Liu. Music databases: Indexing techniques and implementation. In *Proceedings of IEEE International Workshop in Multimedia DBMS*, 1996.

[3] S. Doraisamy and S. M. Rüger. An approach toward a polyphonic music retrieval system. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 187–193, Indiana University, Bloomington, Indiana, October 2001.

[4] M. Dovey. An algorithm for locating polyphonic phrases within a polyphonic piece. In *Proceedings of AISB Symposium on Musical Creativity*, pages 48–53, Edinburgh, April 1999.

[5] J. S. Downie. *Evaluating a Simple Approach to Music Information Retrieval: Conceiving Melodic N-grams as Text*. PhD thesis, University of Western Ontario, Faculty of Information and Media Studies, July 1999.

[6] Poznan University Library, MS 7033, ff. 42-47.

[7] H. H. Hoos, K. Renz, and M. Görg. Guido/mir - an experimental music information retrieval system based on guido music notation. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 41–50, Indiana University, Bloomington, Indiana, October 2001.

[8] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, 1990.

[9] http://www.ecolm.org/.

[10] K. Lemström and J. Tarhio. Searching monophonic patterns within polyphonic sources. In *Proceedings of the RIAO Conference*, volume 2, pages 1261–1278, College of France, Paris, April 2000.

[11] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2001.

[12] B. Pardo and W. Birmingham. Chordal analysis of tonal music. Technical Report CSE-TR-439-01, Electrical Engineering and Computer Science Department, University of Michigan, 2001.

[13] J. Pickens. A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In *Proceedings of the 1st International Symposium for Music Information Retrieval (ISMIR)*, October 2000. See http://ciir.cs.umass.edu/music2000.

[14] J. Pickens. Feature selection for polyphonic music retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Reserach and Development in Information Retrieval*, pages 428–429, September 2001.

[15] J. M. Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, University of Massachusetts Amherst, 1998.

[16] R. E. Prather. Harmonic analysis from the computer representation of a musical score. *Communications of the ACM*, 39(12):119, 1996. See: Virtual Extension Edition of CACM.

[17] W. Rand and W. Birmingham. Statistical analysis in music information retrieval. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 25–26, Indiana University, Bloomington, Indiana, October 2001.