Position Indexing of Adjacent and Concurrent N-Grams for Polyphonic Music Retrieval

Shyamala Doraisamy and Stefan Rüger
Department of Computing, South Kensington Campus
Imperial College London
London SW7 2AZ, England
{sd3,s.rueger}@imperial.ac.uk

Abstract

In this paper we examine the retrieval performance of adjacent and concurrent n-grams generated from polyphonic music data. We deploy a method to index polyphonic music using a word position indexer with the n-gram approach. Using all possible combinations of monophonic sequences from polyphonic music data, "overlaying" word locations within a document are obtained, such as needed with polyphony (i.e. where more than one word can assume the same word position). The feasibility in utilising the position information of polyphonic 'musical words' is investigated using various proximity-based and structured query operators available with text retrieval system. Our experiments show that nested phrase operators improve the retrieval performance and we present the results of our comparative study on a collection of 5456 polyphonic pieces encoded in the MIDI format.

1 Introduction

The robustness of the use of n-grams have been shown in some studies, both text and music (Harding et al, 1997; Doraisamy and Rüger, 2003). In this paper we study the improvement of the n-gram approach to polyphonic music retrieval with the use of term adjacency. In general, proximity information can be quite effective in improving precision of text searches. We outline a method to obtain adjacent and concurrent term locations from polyphonic music. Structured query formulations of music queries are used in investigating the retrieval performance of position indexing of adjacent and concurrent 'musical words'. The following sections contain descriptions of retrieval performance experiments using various structured query formulations.

2 Position indexing

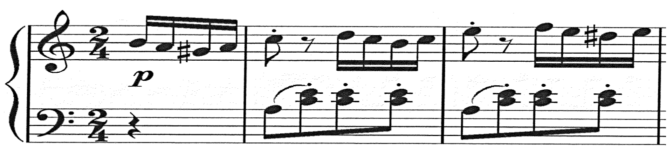
2.1 Pattern extraction and encoding

With the n-gram approach to full-music indexing of polyphonic music data, a "bag of terms" is generated for the indexing process. A polyphonic piece is encoded as an ordered pair of onset times (in milliseconds) and pitch (in MIDI semitone numbers), and these are sorted based on the onsets times. There may possibly be a few different pitches corresponding to one particular onset time. The pitches with similar onset times are grouped together as musical events. Using the gliding window approach, this sequence of events is divided into overlapping subsequences of *n* different adjacent events, each characterised by a unique onset time. For each window, all possible monophonic pitch sequences are extracted and corresponding musical words are constructed.

N-grams are generated from all the possible monophonic sequences using pitch differences and ratios of onset time differences. A sequence of n onset times generates *n*-1 pitch intervals and *n*-2 time ratios, all of which are quantised and encoded as a word with 2*n*-3 letters. For an in-depth discussion, see Doraisamy and Rüger (2003).

2.2 Adjacent and concurrent n-grams

In utilising the time-dependent element of polyphonic music data where the concurrency and sequencing information of the data is considered, a 'polyphonic musical word position indexer' is needed. This is briefly illustrated with the example below. The first five onsets of the music excerpt given in Figure 1 would generate a text document as shown at the bottom of Figure 1, with the corresponding positions of the adjacent and concurrent 'musical words'.



1 bYaYA 2 aYAYl 2 aYAYC 3 AYlFG 3 AYCFh 3 AYlFC 3 AYCFl 4 lFGYJ 4 CFhYJ 4 lFCYN 4 CFlYN 4 lFGY8 4 CFhY8 4 lFCYD 4 CFlYD 4 lFGYd 4 CFhYd 4 lFCY8 4 CFlY8 5 GYJfb 5 hYJfb 5 CYNfb 5 lYNfb 5 GY8fH 5 hY8fH 5 CYDfH 5 lYDfH 5 GYdfL 5 hYdfL 5 CY8fL 5 lY8fL

Figure 1 Adjacent and concurrent 'musical words' with their positions from an excerpt of Mozart's Alla Turca.

2.3 Query formulation

The Lemur Toolkit (http://www2.cs.cmu.edu/~lemur) was modified to index the 'polyphonic musical words'. The structured and proximity-based operators available were used to investigate the retrieval performance. Definitions of operators adopted for our study as given by Lemur are listed in Figure 2.

Sum operator : #sum($T_1…T_n$)

The terms or nodes contained in the sum operator are treated as having equal influence on the final result. The belief values provided by the arguments of the sum are averaged to produce the belief value of the #sum node.

Ordered Distance Operator : #odN ($T_1…T_n$)

The terms within an ODN operator must be found in any order within a window of N words of each other in the text in order to contribute to the document's belief value.

And operator : #and ($T_1…T_n$)

The more terms contained in the AND operator which are found in the document , the higher the belief value of the document.

Or operator: #or ($T_1…T_n$)

One of the terms within the OR operator must be found in a document for that document to get credit for this operator.

Figure 2: Structured and proximity-based operators.

Based on the monophonic study by Pickens (2000), using nested phrase operators in a manner that attempts to recapture the original sequentiality of the songs produces more precise results. We use this query formulation towards our investigation of querying monophonic queries against a polyphonically encoded collection. A monophonic theme extracted from Figure 1 is encoded as:

bYaYA aYAYC AYCIB CIBib BibYa bYaYA aYAYD AYDIA DIAia AiaYa aYaYA aYAYG AYGYb GYbYa bYaYA aYAYB AYBYb BYbYa bYaYA aYAYC

We arbitrarily selected two contiguous musical words for the formation of smaller phrases within a longer query. The query would then be reformulated as:

```
#SUM (   #ODN3(bYaYA aYAYC)
         #ODN3(AYCIB CIBib)
         …
         #ODN3(bYaYA aYAYC))
```

3 Results

A collection of 5456 polyphonic music pieces in the MIDI format and 25 monophonic excerpts extracted from this collection were used as experimental monophonic queries. For each query, there were several performances from the collection that was considered similar. The pitch and rhythm dimensions with the value *n*=4 was used in generating musical n-grams and indexed as PR4, and with position information as PPR4. In reducing the terms indexed, the upper and lower envelopes of the polyphonic data were used and indexed as PR4ENV and PPR4ENV respectively. The probabilistic model supported by Lemur using the Okapi BM25 function for weighting was adopted for retrieval using PR4 and PR4ENV. The relevance assumptions and indexing strategies are discussed in Doraisamy and Rüger (2003).

The precision-at-15 measure was used for evaluation. The weighted average (W.A.) of this precision was used - the retrieval performance of the 25 queries averaged, weighted by the number of relevant documents for each query. Based on the rank position of the relevant document, the Mean Reciprocal Rank (MRR) measure was used. Although normally used with the known-item search, the MRR measure was used in this context based on the best rank of the relevant documents retrieved at precision-at-15. From Table 1, it is clear that the nested phrase operators improve the precision where a MRR measure of 0.70 was obtained.

		W.A.	MRR
PR4		52	0.63
PR4ENV		62	0.66
PPR4	Sum	48	0.57
	Nested	58	0.70
	AND	49	0.56
	OR	62	0.61
PPR4ENV	Sum	62	0.62
	Nested	62	0.66
	AND	56	0.65
	OR	56	0.62

Table 1: Retrieval measures

4 Conclusion and Future Work

We have shown the feasibility of position indexing of polyphonic music and the results show that using nested phrase operators is promising. Although, the improvement in the use of phrase operators has not significantly improved the precision, we continue to work on formulating a similarity measure in retrieving adjacent and concurrent 'polyphonic musical words'.

N-grams generated from monophonic queries would pose a problem when querying against its polyphonic version due to intercepting accompanying onsets. Another problem with the retrieval could also be due to erroneous queries, which are highly probable with music queries such as QBH. A proximity-based operator more specific to music retrieval is clearly needed. We are currently working on a 'musical ordered distance operator' whereby this difference between n-grams generated from the query and the relevant polyphonic document indexed is given by a similarity measure.

Acknowledgements

This work is partially supported by the EPSRC, UK.

References

Doraisamy, S & Rüger, S. (2003). Robust Polyphonic Music Retrieval with N-grams. *Journal of Intelligent Information Systems,* Kluwer Academic Publishers, 21(1), 53–70.

Harding, S.M, Croft, W.B. & Weir, C. (1997), Probabilistic Retrieval of OCR Degraded Text Using N-Grams, *In Research and Advanced Technology for Digital Libraries*, Carol Peters and Constantino Thanos, Editors, pp 345-359.

Pickens, J. (2000). A Comparison of Language Modeling and Probabilistic Text Information Retrieval Approaches to Monophonic Music Retrieval, *ISMIR 2000*, U.S.A.