SEARCHING PAGE-IMAGES OF EARLY MUSIC SCANNED WITH OMR: A SCALABLE SOLUTION USING MINIMAL ABSENT WORDS

Tim Crawford
Goldsmiths, University of London
t.crawford@gold.ac.uk

Golnaz Badkobeh
Goldsmiths, University of London
G.Badkobeh@gold.ac.uk

David Lewis
Oxford eResearch Centre
david.lewis@oerc.ox.ac.uk

ABSTRACT

We define three retrieval tasks requiring efficient search of the musical content of a collection of ~32k page-images of 16th-century music to find: duplicates; pages with the same musical content; pages of related music.

The images are subjected to Optical Music Recognition (OMR), introducing inevitable errors. We encode pages as strings of diatonic pitch intervals, ignoring rests, to reduce the effect of such errors. We extract indices comprising lists of two kinds of 'word'. Approximate matching is done by counting the number of common words between a query page and those in the collection.

The two word-types are (a) normal ngrams and (b) minimal absent words (MAWs). The latter have three important properties for our purpose: they can be built and searched in linear time, the number of MAWs generated tends to be smaller, and they preserve the structure and order of the text, obviating the need for expensive sorting operations.

We show that retrieval performance of MAWs is comparable with ngrams, but with a marked speed improvement. We also show the effect of word length on retrieval. Our results suggest that an index of MAWs of mixed length provides a good method for these tasks which is scalable to larger collections.

1. INTRODUCTION

The historical repertory of Western classical music is increasingly being made publicly available in the form of downloadable (or merely viewable) digital images; these represent pages of the manuscripts or printed books in which they are preserved, and are no different in this respect from other typical online library materials such as texts or maps.

Search facilities within the individual library systems are entirely text-based, usually making use of existing or specially commissioned catalogue data. In a few cases, special viewing interfaces are provided to enhance the user-experience, such as the parallel presentation of multiple part-books on the web-site of the Bayerische Staatsbibliothek in Munich.[1] However, the data markup necessary to achieve this has to be done by human experts, which is impractical in general for large collections.

Musicologists need to be able to browse such collections and to search for specific musical parallels within them; librarians need similar facilities for cataloguing purposes (e.g. to identify unknown or unattributed items). This in turn demands fast search methods of adequate accuracy as a first step in the research process to reduce the number of items needing to be examined more exhaustively.

With very few exceptions, music libraries offer online images rather than encoded scores. Providing the latter involves transcription, which can either be done manually by experts, a time-consuming and expensive process, or automatically by OMR, which inevitably introduces errors of various kinds. As OMR techniques improve in future, these errors are likely to diminish, but highly unlikely to disappear altogether.

For fast searching, we need to extract indexes from the OMR output which enable fast searching at high recall. This depends on the musical data extracted and encoded in the indexes being carefully selected to suit a given use-case. For efficient search of the indexes we can benefit from recent advances in string- and pattern-matching algorithms developed for use in bioinformatics for DNA and protein analysis.

In this paper we focus on three musicologically-motivated user tasks given a corpus of digital images of 16th-century printed music: finding duplicate images within the collection (called *dupl* below); finding pages containing substantially the same music as in a query page (*same*); and identifying pages which have non-identical but related or closely relevant music content, such as in different sections or voice parts than the query (*relv*).

We briefly review earlier work on musical corpus-building, content-based music searching and indexing in section 2. We describe our test collection, relevant aspects of the OMR process and our music indexing strategy in section 3. In section 4, we describe the retrieval tasks and our search method in more detail and our experiments and their evaluation in sections 5 & 6. In section 7 we discuss some of the main findings leading to the proposals for further work in section 8.

1. E.g.: https://stimmbuecher.digitale-sammlungen.de/view?id=bsb00086863

2. PREVIOUS WORK

Corpora of historical music

For musicologists, the amount of historical material available online has exploded in recent years, in line with the general availability of data of all kinds, including audio and video files of performances. This does not mean that their requirements for study and analysis are yet adequately met. The sub-discipline of computational, or digital, musicology tends to devote a great deal effort to data-preparation before the powerful tools of MIR, pattern-matching and statistical analysis can be brought to bear. This is because the majority of the data-resources consist of collections of digital images of the source material, rather than files of its musical content. Traditionally, scores, which attempt to represent the overall musical content of the original documents (which is often, as in the case of the music studied in this paper, distributed between multiple part-books), have been made by human experts; this is inevitably a time-consuming and thus expensive process. The translation (automatic or manual) of musical content from documents or their digital-image surrogates into machine-readable 'texts' is generally referred to as music encoding. While digital tools such as score-editing programs have made this easier, by enabling export to standard formats such as MusicXML[2] and MEI,[3] the process is in general impractically slow for building large collections.

However, there exist some significant and freely available collections of encoded music, such as those maintained by the Center for Computer Assisted Research in the Humanities at Stanford University,[4] which present a wide range of classical music encoded in a number of formats. These encodings permit a variety of ways of searching the data for musical features which are offered by software packages such as Humdrum[5] or Music21.[6]

The online offerings of many digital music libraries in classical music are aggregated in the International Music Score Library Project (IMSLP),[7] adding curated metadata. The resulting meta-collection (almost nine million pages of music) has rapidly become more-or-less indispensable for performers, teachers and students. Searching within IMSLP for most users is done via metadata rather than musical content. An experimental interface for content-based searching, the Peachnote Ngram Viewer,[8] works on the output of commercial OMR software run over a large part of the collection; while this is subject to the significant amount of errors introduced in the OMR process, it powerfully demonstrates the potential of efficient search over a large collection.

In the current work, just as in Peachnote, we are not immediately concerned with an abstract or generalized notion of musical similarity. Rather, we select an encoding that represents the musical feature we wish to match. Our aim is to reduce the search space to a manageable number of musical documents which can be compared or analyzed in more detail manually or by a specialized algorithm. For large collections this task can best be achieved by searching indexes rather than full encodings of each document.

Where the musical features extracted from a document can be represented as some kind of 'text', there are many ways of generating useful indexes which can be searched far more quickly than full texts. These have been the subject of information retrieval research for almost half a century, and provide the mechanisms enabling the almost instaneous search familiar to all who use today's internet. Indexing methods for music – either symbolic or audio – have received less attention, but a number of viable methods have been proposed and/or have found use [1].

For most of the sixteenth and seventeenth centuries, almost all original material comes in the form of separate voice-parts rather than scores. For the purposes of retrieval these can be treated as linear strings of characters depending on the encoding method. There is a vast literature on string-matching, largely motivated by problems from bioinformatics. Some of the resulting, highly-efficient methods have been proposed for music retrieval.

Music retrieval algorithms inspired by bioinformatics

A very recent survey of MIR applications for algorithms developed in bioinformatics research is contained in [2], although this does not include the method adopted in this paper.

The need for pairwise comparison of potentially extremely long, strings representing the structure of molecules such as DNA or proteins, has been addressed by the development of algorithms such as FASTA [3] and its descendants, such as BLAST,[9] which are in common use for DNA analysis. The latter algorithm has found musical uses in the audio [4] and symbolic [5] domains. BLAST has also found use in recent work on audio cover-song recognition in [6], where the major speedup in retrieval it brought was found to compensate for a slight degradation in retrieval accuracy. Most recently, [7] reports on the application to music of methods originally designed for bioinformatics. These include multiple sequence-alignment methods such as MAFFT [8].

The present work uses a method which is finding increasing acceptance within bioinformatics, but has not, as far as the authors are aware, previously been applied to music: minimal absent words (MAWs). Here we briefly introduce the concept.

A word is an absent word of a sequence if it does not occur in the sequence. An absent word is minimal if all its proper factors occur in the sequence. Absent words are negative information about the sequence. These objects have been extensively studied in combinatorics on words and it is known that although the number of absent words of a sequence is exponential with respect to the size of the sequence, the number of minimal absent words is only linear with respect to the length of the sequence [9].

2. http://www.musicxml.com
3. http://music-encoding.org
4. http://www.musedata.org and http://kern.ccarh.org
5. http://humdrum.ccarh.org
6. http://web.mit.edu/music21/
7. http://imslp.org
8. http://www.peachnote.com
9. Basic Local Alignment Search Tool; http://blast.ncbi.nlm.nih.gov/

Crochemore et al. in [10] presented a linear-time algorithm to compare two documents by considering all their minimal absent words, using a length-weighted index measure.

In recent years, the significance of minimal absent words has been studied in several biological studies. In [11] absent words for four human genomes were computed, and it was shown that intra-species variations in minimal absent words were lower than inter-species variations.

Furthermore, minimal absent words have been exploited for building phylogenies [12], for measuring dissimilarities/similarities between bio-sequences [13] and for many other applications [14].

3. TEST COLLECTION & OMR

The collection consists of 31,721 page-images of 16th-century printed music, which have all been subjected to OMR. The music was scanned from archival microfilms, in which the several individual part-books for a given item (usually four but up to as many as 12), one for each voice, follow in sequence as preserved under their single shelfmark. In almost every case they show two facing pages in a single image; these were each separated by us into two single page-images.

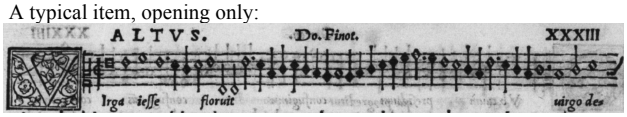
The collection has associated metadata which gives bibliographical information for each book, but not to the level of musical items; so, for example, while the general sequence of musical items in each book is listed, with titles and original composer ascriptions, the locations of items in the part-books is not recorded. Thus, it is in general impossible to associate automatically a page image with the music on it. This provides the motivation for the present work, aimed at designing a finding aid for researchers or librarians wishing to identify similar or related music within the collection.

The OMR tool we use is Aruspix, a program specifically designed for early printed music.[10] While this represents the current state of the art for this repertory [15], recently reported work suggests that significant progress is possible in the near future [16]. However, it is unlikely that 100% accuracy in OMR will ever be consistently achieved for any repertory; for this reason we maintain that fast, error-robust search methods will always be in demand. Aruspix saves its recognized output as MEI (mensural)[11] from which we can extract various kinds of musical sequence. (See Fig 1.)

Typical errors made by OMR systems can be of duration (wrong/missing time-signatures; wrong/missing note-values) and of pitch (wrong/missing clefs; wrong/missing key-signatures; wrong/missing accidentals). The vertical location of symbols such as note-heads on the staff is usually recognized securely; this corresponds to diatonic pitch. Changes of clef tend to compound this effect as pitches are affected over a span of notes (usually until the next line of music), so it is helpful to use relative pitches, i.e. intervals. We have found sequences of diatonic intervals to be the most useful for our purposes.

We generate a single diatonic-interval string for each page using a simple alphabetic code devised by RISM[12] for rapid searching of musical incipits (See Figure 1). Letters in upper case represent ascending intervals, lower case descending; same note is indicated by a hyphen.[17]

A typical item, opening only:



MEI output from Aruspix (opening only, simplified):

```
<clef line="3" shape="C" />
<mensur sign="C" slash="1" />
<note pname="e" oct="4" dur="brevis" />
<note pname="d" oct="4" dur="semibrevis" lig="recta" />
<note pname="f" oct="4" dur="semibrevis" />
<note pname="e" oct="4" dur="semibrevis" />
<dot ploc="f" oloc="4" />
<note pname="d" oct="4" dur="semiminima" />
<note pname="c" oct="4" dur="semiminima" />
<note pname="d" oct="4" dur="minima" />
<custos pname="c" oct="4" />  [Spurious:Note missing!]
<note pname="e" oct="3" dur="minima" />
<note pname="e" oct="3" dur="minima" />
<note pname="e" oct="4" dur="minima" />
```

(NB Because the clef has been mis-recognized, all pitches are a third too low; also, in line 11, a note has been mis-read as a *custos*.)

Diatonic pitch sequence:

MEI:	e4	d4	f4	e4	d4	c4	d4		e3	e3	e4
Correct:	g4	g4	a4	g4	f4	e4	f4	g4	g3	g3	g4

Diatonic interval sequence:

MEI:	-1	+2	-1	-1	-1	+2	-7		0	+7
Correct:	0	+1	-1	-1	-1	+1	+1	-8	0	+8

Encoded diatonic interval sequence:

MEI:	a	B	a	a	a	B	f		-	G
Correct:	-	A	a	a	a	A	A	g	-	G

Figure 1. The (erroneous) MEI output from Aruspix, and the correct encoding, for a typical item (opening only),[13] and the sequences we derive from it.

4. TASKS AND METHOD

The three tasks we approach are to recognise page-images which: (a) are duplicates (i.e. different shots/scans of the same page); (b) contain substantially the same music (which may be distributed differently across adjacent pages); (c) contain related but not identical music (this may be from a different voice-part, from a different section of the same piece, or from a derivative work).

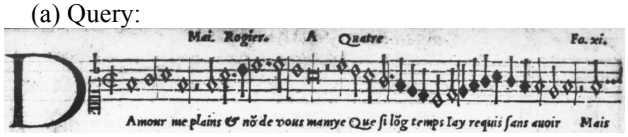
Task (a) involves finding near-identical matches; however, the OMR output, and hence the indexes we extract, are not necessarily exactly the same, owing to recognition errors or small differences in photographic conditions, etc. For task (b), although in principle the encodings on which we base our searches should be largely identical, we cannot be sure that each page of different editions of a pieces of music has exactly the same content; often, the page layout is different, or the music is distributed over multiple pages in one or other copy. Furthermore, there

10. http://www.aruspix.net
11. http://music-encoding.org/schema/2.1.1/mei-Mensural.rng
12. *Répértoire Internationale des Sources Musicales*; see http://www.rism.info/home.html
13 D. Phinot (c.1510-c.1555), Altus part of 'Virga Jesse floruit', from *Primus liber cum quatuor vocibus : Mottetti del frutto a quarto* (Venice: Gardane, 1539)

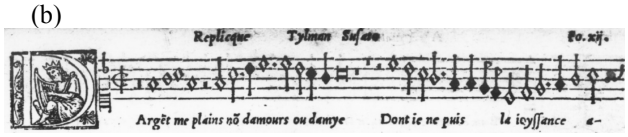
may be extraneous material 'foreign' to the query page printed on the same page at the beginning or the end of the piece in question.

The 'related music' category is best illustrated by example; all of the following were found as high-ranking matches to the query page (2a) using our methods despite the fact that they tend to diverge after a statement of the opening motif:

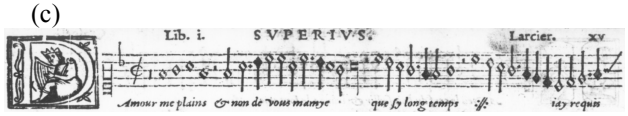
(a) Query:



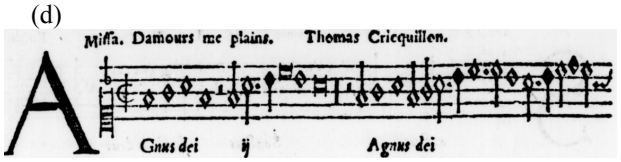
(b)



(c)



(d)



(e)

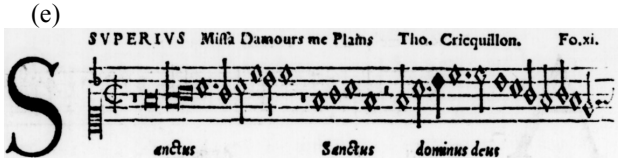


Figure 2. Examples of music 'related' to a query (a). (N.B. These matches were based on full pages of music, not just on the incipits displayed here.)

Our query page (2a) was the Superius part of 'D'amours me plains', a *chanson* by Maistre Rogier.[14] The following item in the same book is a *replicque*, or response, to the *chanson*, with a different text, by Tylman Susato, based on the same musical motifs; this was ranked second. Another piece based on Rogier's *chanson*, this time with the same text, by Larcier was in fact ranked first. The third-ranked item was the 'Agnus Dei' from Thomas Crequillon's parody mass on the song, *Missa Damours me plains*; the 'Sanctus' from the same mass was ranked in fourth place.

Further examples of 'related' music might include separate sections of a work, or arrangements with completely different texts which were catalogued as separate items. In fact, in early testing of our method, we discovered that the *Recercar Undecimo* by an unidentified composer in a 1593 miscellany,[15] is in fact a previously unrecognized instrumental arrangement of a motet, 'In die tribulationis,' by 'Damianus', probably Damien Havericq (active 1538-56), published half a century earlier in 1549.[16]

At first we extracted ngrams from the page-encodings, i.e. fixed-length substrings of length k extracted sequentially starting at each character in the string in turn. These were built into a trie (suffix-tree) structure for efficient searching. We then counted the number of ngrams in common between the query and each page of the collection in turn. Although this worked well enough for task (a), we encountered difficulties with tasks (b) and (c) for two reasons: firstly, this naïve ranking did not take account of the fact that longer pages are more likely to contain ngrams which appear in the query by chance, and secondly, we were ignoring the order of locations of the ngrams, which should be the same in query and target documents, for obvious musical reasons.

The first difficulty was overcome by using Jaccard distance[17] rather than a raw count of coincident ngrams; all results reported here use this measure as a basis for search-result ranking. The second problem can be tackled by including ngram-location in the index and sorting the array of results. However, the process of ensuring an ordered match from the ngram set adds undesirable computational complexity.

Turning to a method that has found wide acceptance in recent bioinformatics, we used minimal absent words (MAWs)[18] instead of ngrams. We have found this to be highly successful, both in terms of the reduction of the amount of data that has to be searched and because of the fact that MAWs retain the order and structure of the original document, avoiding the necessity for the secondary expensive sorting routine.

5. EXPERIMENTS

For the purposes of the comparison between retrieval using ngrams and MAWs, we ran experiments based on the three user tasks outlined above using a version of the software implemented in Javascript on a MacBook Pro (2.5 GHz Intel Core i7 with 8GB RAM), running OS X 10.13.3. The software was run in a standard web browser (Safari) via localhost. While we would not consider this to be a sensible setup for production work, it had the advantage of not requiring network access with consequent latency issues.[19]

For each task we ran the searches using indexes of different word-lengths (3-10 characters) and the two word-types (*ngrams* and *MAWs*). In addition (as explained below) we used an index of MAWs of mixed length (4-8 characters).

For ngrams we did not include the result-sorting routine. We expect that sorted ngram results will give the overall best retrieval performance, but this will come at a significant cost in terms of speed, not evaluated here. In

14 Premier livre des cha[n]so[n]s a quatre parties (Antwerp: Susato, 1543, f. xi

15 *Fantasie recercari et contrapunti a tre voci* (Venice: Gardane, 1593)

16 *Libro secondo de li motetti a tre voce da diversi* (Venice: Scotto, 1549), item XVIII

17. https://en.wikipedia.org/wiki/Jaccard_index

18. http://www.lix.polytechnique.fr/SeminaireDoctorants/AliceHeliouMotsAbsents.pdf

19. The code and encoded data are accessible at: http://doc.gold.ac.uk/~mas01tc/ISMIR2018/

fact, we believe that these will find their best use on reduced result lists after the initial indexed search.

Before each experiment, the appropriate full index needs to be loaded into a trie (suffix-tree) structure. This process can take up to a minute or so for the larger indexes which also use a lot of memory. Index loading is not considered as part of our experiments, since the indexes would need to be retained as a persistent service (probably distributed between machines) in a production system.

Each task has its associated query-list derived from the predetermined ground truth (see below). These contain different numbers of queries (48, 107 and 334 for the *dupl*, *relv* and *same* tasks, respectively). Each query, consisting of a set of index words, was run by searching in turn for each word on the complete index, counting the words in common between query and target pages, with results sorted by Jaccard distance. Where the number of common words was less than 6 the search was regarded as unsuccessful and no results were returned.[20] For certain word lengths, no MAWs were generated for some pages (see Discussion, below); for these cases, too, no results were returned.

6. EVALUATION

We had previously gathered ground truth using a web-interface allowing a user to annotate documents in ranked results as (a) a duplicate image of the same page (*dupl*); (b) a page containing substantially the same music (*same*); or (c) related or relevant music, such as that belonging to a different voice-part or section of a work (*relv*).

In the three graphs that follow we present the average rank at which known matches from the ground truth lists for a given word length were retrieved in the three experiments. Since we were mainly interested in high-ranking matches, we gave all items falling beneath the rank of 20 a uniform rank of 25.

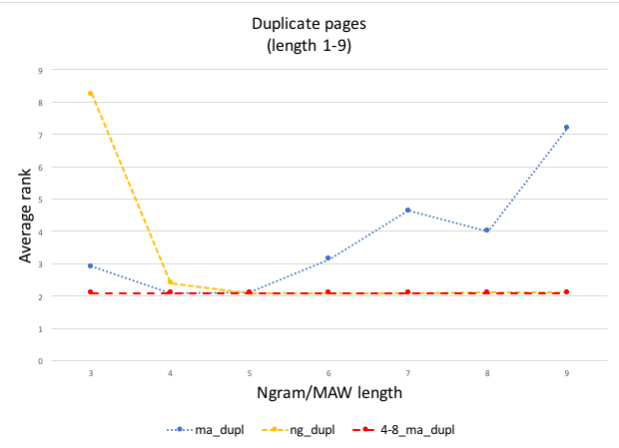


Figure 3. Average ranks for matches of ‘duplicate’ pages.

In our experiments with our test dataset, retrieval performance for the *dupl* task was found to be similar for ngrams and MAWs of length 5 characters. We do not expect, however, that this will remain true for all other collections, and it is not the case for the other tasks. For this reason, we also performed all the tasks with a mixed-length index of MAWs (4-8 characters) which gave results almost identical to ngrams in the *dupl* task, consistently high in the ranked results in the case of the *same* task, and the overall best for the *relv* task.

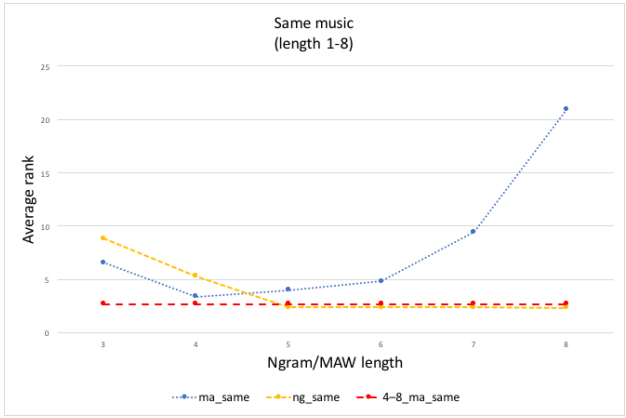


Figure 4. Average ranks for matches of ‘same music’ pages

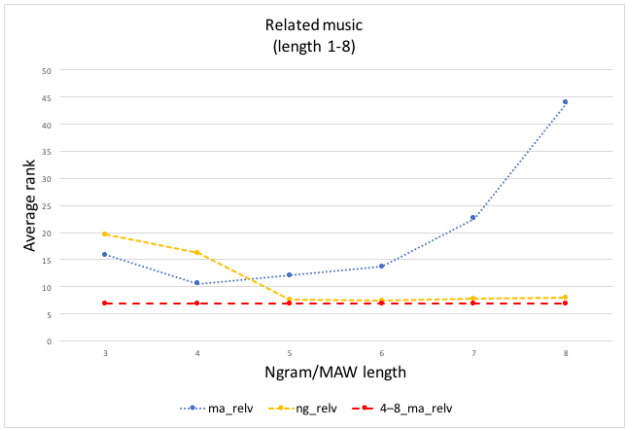


Figure 5. Average ranks for matches of ‘related music’

The experiments are named using ‘ng’ and ‘ma’ to indicate the use of ngrams or MAWs. The dashed lines on the graphs represent the average rank for the searches using mixed-length MAWs (4-8 chars); these are not quite as good as the best results for ngrams, but very close, and the speed is much faster.

7. DISCUSSION

The usefulness of MAWs is highly data-dependent. Over a length-range of 3 to 10 characters, the number of MAWs generated for each page, while lower than the number of ngrams of those lengths, falls off in a way that means that there is simply not enough data for consistent recognition beyond a certain length.

20. This arbitrary number was arrived at in early testing as lower numbers gave essentially useless results.

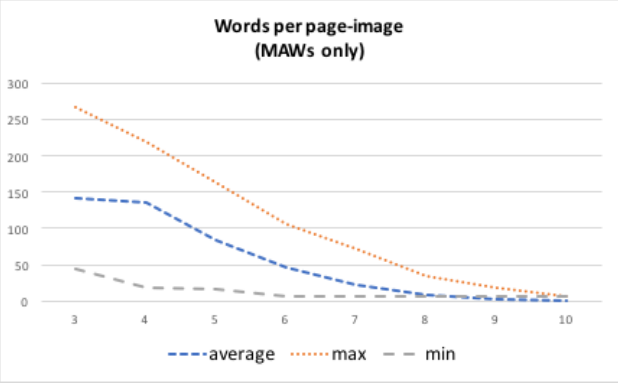


Figure 6. Number of minimal absent words per page-image (average, maximum and minimum)

However, a database of MAWs with mixed lengths (4-8) always produces enough data for matching and performs almost as well as the best ngram length, but is much faster in operation.

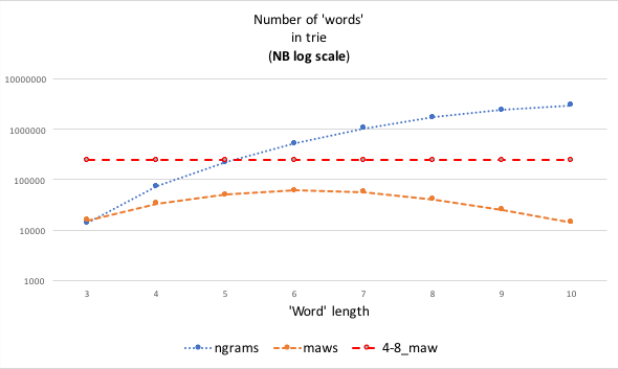


Figure 7. Number of words in the trie structure for the entire collection. NB Log scale!

Finally, we show the average search time per word in the collection for each word-length:

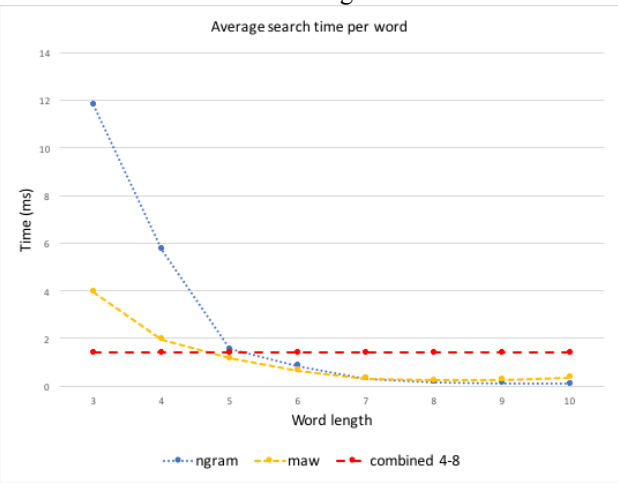


Figure 8. Average search time (ms) per word for each word length

Bearing in mind that we need to search for each word in the query page in turn, it can be seen that the lower numbers per page of MAWs compared to ngrams, and the consequently smaller index size brings a significant speed advantage. This is particularly important in a search tool which is to be operated by human researchers, who increasingly expect retrieval response comparable to that encountered in everyday web searching.

A possibly interesting finding, whose significance needs further investigation, is that there is a fairly consistent range of ngram and MAW lengths (viz. roughly between 4 and 8 characters) that produces useful results - this may relate to the nature of the musical data, i.e. to the 'language' or style of the music, but this needs to be tested formally with a range of different repertories.

8. FURTHER WORK

In future work, we intend to compare the efficacy and performance of MAWs with standard algorithms such as BLAST.

Since our use of ngrams in this research was to provide a benchmark for the efficacy of MAWs, limited attempts have been made to optimise them for retrieval speed. We are confident that with the data that we now have on the most effective ngram lengths, effort can be put into algorithmic efficiency for a comparison of the two technologies based on their real-world speed.

In many retrieval tasks, it is sufficient simply to return a ranked list of the k best matches for a query, but in the tasks we investigate here, there is an approximately binary relevance judgement to be made. The number of relevant documents can vary from 0 to over 100, so finding an appropriate thresholding value is important. Statistical approaches to thresholding have proved useful in the high-dimensional spaces associated with audio searching [18], and this is a sine qua non in text retrieval.

We intend to increase the size of our test collection to investigate how well it scales. In order to achieve this, we hope to establish a consortium of international music libraries to contribute images and metadata, with the ultimate goal of providing a comprehensive search tool for musicologists. This requires further work on system architecture and management of distributed data and processing.

In principle, there is no reason why similar techniques could not be used on other monophonic repertories, and we hope to widen the scope of our work through our continuing association with projects such as SIMSSA21 and TROMPA.[22]

MAWs present a valuable new method for music research which is scalable to collections a good deal bigger than our test set of 32k pages. The technique is generally applicable to any repertory which is reducible to monophonic parts or streams, allowing fast approximate retrieval of large queries over web-scale collections of noisy data.

21. *Single Interface for Music Score Searching and Analysis* (project funded by Social Sciences and Humanities Research Council, Canada)
22. *Towards Richer Online Music Public-domain Archives* (Horizon 2020 project funded by the EU, 2018-21)

9. ACKNOWLEDGMENTS

We gratefully acknowledge the help and advice of Prof. Maxime Crochemore and Dr Jeremy Pickens in the writing of this paper. Our thanks are due to Solon Pissis for help with adapting his MAW-extraction code for our musical purposes. The work was partially funded by the UK AHRC project, Transforming Musicology, AH/L006820/1.

10. REFERENCES

[1] M. Schedl, E. Gómez and J. Urbano: "Music Information Retrieval: Recent Developments and Applications," *Foundations and Trends in Information Retrieval*, Vol. 8, No. 2-3 127–261, 2014.

[2] D. Bountouridis: "Music Information Retrieval Using Biologically-Inspired Techniques," PhD dissertation, Utrecht University, 2018.

[3] W. R. Pearson and D. J. Lipman: "Improved tools for biological sequence comparison", *Proc Natl Acad Sci USA*. 85(8), 2444-8, April 1988.

[4] R. B. Dannenberg and N. Hu: "Pattern Discovery Techniques for Music Audio", *Proceedings of the International Symposium on Music Information Retrieval*, 2002.

[5] J. Kilian and H. Hoos: "MusicBLAST — Gapped Sequence Alignment for MIR", *Proceedings of the International Symposium on Music Information Retrieval*, 2004.

[6] B. Martin, D.G. Brown, P. Hanna and P. Ferraro: "BLAST for Audio Sequences Alignment: A Fast Scalable Cover Identification Tool," *Proceedings of the International Symposium on Music Information Retrieval*, 529-534, 2012.

[7] D. Bountouridis, D.G Brown, F. Wiering and R.C. Veltkamp: "Melodic similarity and applications using biologically-inspired techniques", *Applied Sciences*, Special Issue on Sound and Music Computing, 7. 12, 2017.

[8] K. Katoh, K. Misawa, K. Kuma, and T. Miyataa: "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," *Nucleic Acids Res.* 15; 30 (14), 3059-66, July 2002.

[9] M. Beal, F. Mignosi, A. Restivo and M. Sciortino: "Forbidden words in symbolic dynamics," *Advances in Applied Mathematics*, Vol 25(2), 163–193, 2000.

[10] M. Crochemore, G. Fici, R. Mercas and S. Pissis: "Linear-Time Sequence Comparison Using Minimal Absent Words & Applications," *Proceedings of the Latin American Theoretical Informatics Symposium (LATIN)*, 334-346, 2016.

[11] S.P. Garcia and A.J. Pinho: "Minimal absent words in four human genome assemblies," *PLOS ONE*, Vol. 6 (12), 2011.

[12] S. Chairungsee and M. Crochemore: "Using minimal absent words to build phylogeny," *Theoretical Computer Science*, Vol. 450, 109–116, 2012.

[13] C. Barton, A. Heliou, L. Mouchard and S.P. Pissis: "Linear-time computation of minimal absent words using suffix array," *BMC Bioinformatics* Vol. 15 (1), 2014.

[14] W.K. Sung, *Algorithms in Bioinformatics: A Practical Introduction*, CRC Press, London, UK, 2009.

[15] L. Pugin and T. Crawford, "Evaluating OMR on the Early Music Online Collection," *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 439–44, 2013.

[16] J. Calvo-Zaragoza, J.J. Valero-Mas and A. Pertusa: "End-to-end Optical Music Recognition Using Neural Networks," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 472–477, 2017.

[17] J. Diet and M. Gerritsen: "Encoding, Searching, and Displaying Music Incipits in the RISM-OPAC," *Music Encoding Conference 2013*, Mainz, Germany (unpublished).

[18] M. Casey, M. Slaney and C. Rhodes: "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16 (5), 1015-1028, 2008.