

Stravinski/De Montfort University at the MediaEval 2014 C@merata Task

Tom Collins
Faculty of Technology
De Montfort University
Leicester, UK
+44 116 207 6192
tom.collins@dmu.ac.uk

ABSTRACT

An overview is provided of the Stravinski-Jun2014 algorithm and its performance on the MediaEval 2014 C@merata Task. Stravinski stands for STaff Representation Analysed VIa Natural language String Query Input. The algorithm parses a symbolic representation of a piece of music as well as a query string consisting of a natural language expression, and identifies where event(s) specified by the query occur in the music. The output for any given query is a list of time windows corresponding to the locations of relevant events. To evaluate the algorithm, its output time windows are compared with those specified by music experts for the same query-piece combinations. In an evaluation consisting of twenty pieces and 200 questions, Stravinski-Jun2014 had recall .91 and precision .46 at the measure level, and recall .87 and precision .44 at the beat level. Important potential applications of this work in music-educational software and musicological research are discussed.

1. INTRODUCTION

Given a natural language query and a piece of music in digital staff notation representation, the C@merata task [6] evaluates an algorithm's ability to identify where one or more events specified by the query occur in the music. It is the latest example of a long-standing interest in querying music represented as (or derived from) staff notation. The C@merata task challenges researchers to extend current knowledge in two respects:

1. Accepting a music-analytic query in the form of a natural-language string, such as "perfect fifth followed by a D4";
2. Reliably retrieving instances of higher-level music-theoretic concepts from staff notation, such as functional harmonies (e.g., "Ib") or cadences (e.g., "interrupted cadence").

One application of an algorithm that performs well on the C@merata task would be within music notation software, so that students could query and hear/see results for the pieces with which they are working, in order to develop their understanding of various music-theoretic terms.

2. APPROACH

2.1 Overview

The Stravinski-Jun2014 algorithm that was entered in the C@merata task is embedded in a Common Lisp package called MCStylistic-Jun2014 (hereafter, MCStylistic), which has been under development since 2008 [1].¹ MCStylistic includes

¹ <http://www.tomcollinsresearch.net>

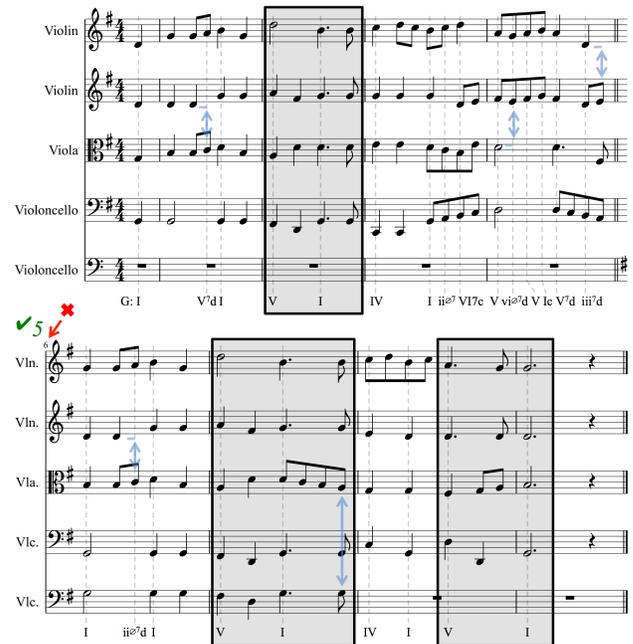


Figure 1. Prelude to *Te Deum* H146 by Marc-Antoine Charpentier (1643-1704), annotated with a bar number error (tick and cross), intervals of a harmonic second (arrows), functional harmonies below each staff, and three perfect cadences (black boxes).

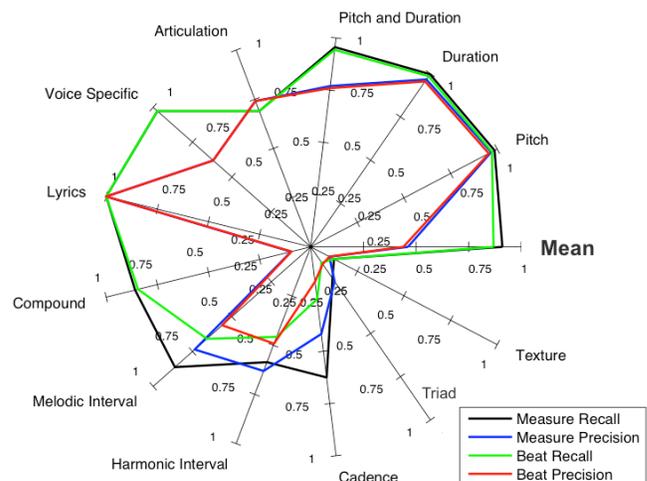


Figure 2. Results of the Stravinski-Jun2014 algorithm on the MediaEval 2014 C@merata task. Overall results are indicated by the mean label, and followed by results for twelve question categories.

implementations of algorithms from the fields of music information retrieval (MIR) and music psychology [2-5].

From a natural language perspective, there are two types of queries: compound queries such as “a Bb followed a bar later by a C followed by a tonic triad”, and ordinary queries such as “perfect cadence”. Stravinsqi checks the query string for compound queries and splits it into N query elements if necessary, e.g., “a Bb” and “a bar later by a C” and “tonic triad”.

The piece is converted from its MusicXML format to kern format using the `xml2hum` script.² The kern file is parsed by import functions in `MCStylistic` to give the following representations, which are referred to as *point sets*: (1) instrument/staff and clef names at the beginning of each staff; (2) bar numbers where time signatures are specified, together with the number of beats per bar, the type of beat, and the corresponding ontime (incrementing time in staff notation); (3) a point-set representation of the piece, where each point represents a note. The five-dimensional point consists of the ontime of the note, its MIDI note number, its morphetic pitch number [3], its duration in crotchet beats, and its numeric staff number; (4) a point-set representation of the piece with three extra dimensions, one each for articulation, dynamics, and lyrics information; (5) a point-set representation of the piece, where each point represents a notated rest.

Each query element is passed to several sub-functions (e.g., `harmonic-interval-of-a`, `duration&pitch-time-intervals`, `rest-duration-time-intervals`, etc.), along with the appropriate point set(s). For example, the function `rest-duration-time-intervals` takes a query element, the point set of notated rests, and the point set of instrument/staff and clef names as its arguments, because these three information sources are sufficient for locating rests of specific duration. If a query string is ordinary (contains one element only), then the time windows in the first nonempty sub-function’s output are passed to a final function that converts the time windows into the XML format required by the task. For compound queries, plausible sequences of time windows for the component query elements are merged before passing to the final syntax-conversion function.

2.2 Example Output for Three Sub-Functions

Figure 1 contains example output for the sub-functions `harmonic-interval-of-a`, (arrows indicate retrieved harmonic seconds), `HarmAn->Roman` (functional harmonic labels below each staff), and `cadence-time-intervals` (three perfect cadences surrounded by black boxes). All three functions involve implementing and extending MIR/music-psychology algorithms to achieve promising results, especially for the higher-level music-theoretical concepts such as functional harmonies and cadences.

3. RESULTS AND DISCUSSION

Figure 2 shows recall and precision results for the Stravinsqi algorithm on the 2014 C@merata task. The *measure* metrics reward an algorithm’s output if it is in the same bar/measure as a ground-truth item, whereas the *beat* metrics require an algorithm’s output to be in the same bar *and on the same beat* as a ground-truth item. The mean category in Figure 4 shows the overall

results, with Stravinsqi having recall .91 and precision .46 at the measure level, and recall .87 and precision .44 at the beat level.³

Stravinsqi’s strong performance on the first eight of twelve categories (pitch, duration, ..., melodic interval) is encouraging, as is the small decrease in recall (.91 to .87) and precision (.46 to .44) with the change from measure- to beat-level granularity. The drop in precision for compound queries is due to over-lenient criteria used to select and combine time intervals for the different elements that comprise a compound query. This can be fixed in future work. For triad labelling, Stravinsqi suffered from an under-labelling issue in two instances, missing two first-inversion triads because the same triad in root position preceded them, and the two triads got one root-position label. The triad and texture categories are somewhat underrepresented in the training and test data, and so more attention ought to be given to these categories in future. Less-than-perfect performance on melodic and harmonic interval questions can be attributed to inconsistencies between the task description/training collection, and test collection.

4. CONCLUSION

Algorithms that perform strongly on the C@merata task open up new, interesting potential applications in music education and musicological research. The Stravinsqi algorithm described above is one such strong performer, and has effectively solved seven of the twelve C@merata task categories shown in Figure 4 (pitch, duration, pitch and duration, articulation, voice specific, lyrics, and melodic interval). As for the remaining five categories, future work will involve bug fixes, resolving task inconsistencies, and acquiring more data for cadence and texture query categories. It may also be helpful to have two experts provide annotations for the higher-level music-theoretic concepts. The addition of new, higher-level music-theoretic query categories would be welcome in future iterations of C@merata as well, in order to keep the task at the forefront of research in music computing.

5. REFERENCES

- [1] Collins, T. 2011. *Improved Methods for Pattern Discovery in Music, with Applications in Automated Stylistic Composition*. Doctoral Thesis. Faculty of Mathematics, Computing and Technology, The Open University.
- [2] Krumhansl, C. L. 1990. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, NY.
- [3] Meredith, D., Lemström, K., and Wiggins, G. A. 2002. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *J. New Music Res.* 31, 4, 321-345.
- [4] Pardo, B., and Birmingham, W. P. 2002. Algorithms for chordal analysis. *Comput. Music J.* 26, 2, 27-49.
- [5] Sapp, C. S. 2005. Visual hierarchical key analysis. *ACM Computers in Entertainment.* 3, 4, 1-19.
- [6] Sutcliffe, R., Crawford, T., Fox, C., Root, D. L., and Hovy, E. 2014. The C@merata task at MediaEval 2014: natural language queries on classical music scores. In *MediaEval 2014 Workshop*, Barcelona, Spain, October 16-17 2014.

² <http://extras.humdrum.org/bin/osxintel64/>

³ Stravinsqi is labelled DMUN03 in the task overview paper [6]. The other runs DMUN01 and DMUN02 are not remarkable: incorrect bar numbers in four pieces (see, for instance, the cross and correction in Figure 1) and `xml2hum` conversion caused issues in DMUN01 and DMUN02.