

# TCSL at the MediaEval 2014 C@merata Task

Nikhil Kini

Tata Consultancy Services Ltd.  
Innovation Labs, Thane  
nikhil.kini@tcs.com

## ABSTRACT

We describe a system to address the MediaEval 2014 C@merata task of natural language queries on classical music scores. Our system first tokenizes the question to tag the musically relevant features in the question using pattern matching. In this stage suitable word replacements are made in the question based on a list of synonyms. Using the tokenized sentence we infer the question type using a set of handwritten rules. We then search the input music score based on the question type to find the musical features requested. MIT's music21 library [2] is used for indexing, accessing and traversing the score.

## 1. INTRODUCTION

To those interested in studying music scores, and especially in the field of Musicology, it is often necessary to search for, or refer to, particular sections of a music score that represent relevant musical features, just as a data scientist might look for interesting patterns in an abundance of data. Manually going through the score is prone to inefficiency, oversight, and requires the expert knowledge of actually understanding the score. This paper aims to develop specifications for tools that automate the task of search and retrieval of musical passages from a score with natural language queries. The problem may be defined as: given a computer representation of music as a score in a particular format (in our case, musicXML), and given a short English noun phrase referring to musical features in the score, search and list the location of all occurrences of the said musical features in the score. A complete description of the task can be found at [1].

While a large body of work is available in other domains for Natural language understanding, as well as for searching through sheet music scores, we did not come across any work that combines these aspects. A natural language understanding systems survey may be found in [3], and work done in non-trivial search on sheet music scores can be found in [4], [5], [6], [7].

## 2. APPROACH

Figure 1 presents the main modules of our system. Since we treat the problem as one of natural language understanding (of the question) and searching (through the musicXML), we define a set of question classes based on the searchable musical features, and propose a specific search method for each type of question. The main operations performed by our system are as follows:

### 2.1 Identifying tokens in the question

In the tokenizing step, words representing musically important features are marked/tokenized. We use 3 or 4 letter markers for the token class. After tokenization, the sentence will contain tokens grouped with the value of the token, each token-value pair grouped by parentheses, and the token and the value will be

separated by a comma. For example, "quarter note then half note then quarter note in the tenor voice" is output as "(DUR, quarter note) (SEQ, then) (DUR, half note) (SEQ, then) (DUR, quarter note) in the (PRT, tenor voice)". Another example is "melodic octave" becomes "(HRML, melodic) (INT, octave)".

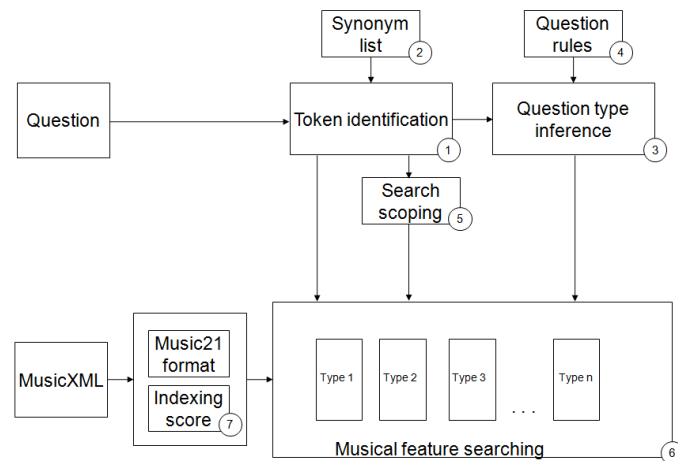


Figure 1. System for natural language sheet music querying

### 2.2 Synonyms List

A list of synonyms is referred to during tokenizing for substituting words that refer to the same feature. This serves two purposes: 1) to cover all manners of asking for the same feature and 2) standardizing the different ways of asking for the same thing so that specifying the subsequent modules becomes simpler. The list of synonyms can be updated as new ways of asking the same feature are discovered when users actually query the system.

### 2.3 Inferring the question type

The tokenized output (with synonym list substitution) is the input to the module which infers the question type. A handcrafted set of rules was used to guess what type of question is asked based on the constituent tokens (see section 2.4). Looking at all questions available to us so far - task description, training set, test set - we specify the following types of questions: simple note, note with expression, interval (harmonic), interval (melodic), lyrics, extrema (highest or lowest note), time signature, key signature, cadence, triads, texture, bar with dynamics, consecutive notes, combination of the above.

### 2.4 Question rules

Based on the tokens present in the question phrase, we can write rules to guess the type of the question. For simple questions made up of only one question this is straightforward. For the phrases which contain a combination of elementary question types, some parsing capability might be necessary. We will address this in future work.

## 2.5 Search scope

An important part to getting the right answer is limiting the search scope. For example, in the question "A sharp in the Treble clef", we are not just looking for any A#, but particularly in the Treble clef. Our tokens PRT and CLF can be used to scope the search. We look only in these parts during searching or we filter only those search results as answers which are within this search scope.

## 2.6 Searching for the answer

The last step is searching the musicXML score for the identified token/token combination. This step is still a work in progress. We make extensive use of music21 capabilities.

## 2.7 Score index

This is a list of all the notes in the score stored with the following associated information for each note: note name, note letter, accidental, pitch class, note octave, bar, offset, note length, part number, part id and whether this is a rest or a note. (This terminology is as defined in music21).

## 3. RESULTS AND DISCUSSION

Upon release of the results, we saw that the organizers had also used a scheme of classification for the questions. Reconciling the organizers' and our question types, we saw that as far as the test questions go, we had all possibilities covered.

Table 1. Test set results

#	Question type	Beat		Measure	
		P	R	P	R
1	simple_length	0.979	0.988	0.991	1
2	simple_pitch	0.959	0.963	0.982	0.986
3	pitch_and_length	0.723	0.892	0.754	0.93
4	stave_spec	0.661	0.987	0.661	0.987
5	melodic_interval	0.894	0.683	0.904	0.691
6	followed_by	0.733	0.688	0.842	0.789
7	word_spec	0.261	1	0.261	1
8	perf_spec	0.066	0.897	0.066	0.897
9	harmonic_interval	0	0	0	0
10	cadence_spec	0	0	0	0
11	triad_spec	0	0	0	0
12	texture_spec	0	0	0	0
13	all	0.633	0.821	0.652	0.845

Our classes corresponding to the organizers' "Question type" in Table 1 are: 1, 2, 3 - simple note, 4 - simple note with expression, 5 - simple note with a staff scope, 6 - simple note with a lyrics scope, 7 - consecutive notes, 8 - interval (melodic), 9 - interval (harmonic), 10 - cadence, 11 - triad, 12 - texture.

Table 1 shows beat and measure precision recall scores for results produced by our system for the test set. The strongest performance is seen in the 'simple notes' category (simple pitch, simple length, pitch and length). This is no surprise as these question phrases are the easiest to handle. *Perf\_spec* questions are simple note with expression type questions (involving for example, mordant and trill). *Word\_spec* is pitch/length occurring over a certain word in the lyrics. Although these were not handled by our implementation, some results were returned because the system fell back to the simple note type, which explains the non-zero precision and recall. For e.g. "F trill" returns all F notes. *Followed\_by* is equivalent to consecutive notes. *Melodic\_interval*

is a type with our system too, and the system performs decently on both these types.

Although search was not implemented for *harmonic\_interval*, *cadence\_spec*, *triad\_spec* and *texture\_spec*, nearly all questions for these types were correctly classified by our system. No answers were returned for these types of questions, which results in the zero scores seen in the table. Only 8 questions remained unclassified in the test data.

## 4. CONCLUSION

The system implemented based on the specifications in this paper performs decently on single musical feature retrieval. A study of the errors in this implementation might even be able to take the precision and recall for such simple types to 1, and this will be the aim of the next cycle of development.

While our system performs well on the simple question phrases, the more complex question phrases still need work. As a question grows more complicated to include multiple musical features, we will need to evolve a more complex parsing strategy to identify questions. It is possible that the specification of the system will need to be revisited to take into account all the possibilities.

The scope of the system specification is limited mainly to what we have observed in the task description and the training set, and these are in no way exhaustive of the types of queries that can be asked.

## 5. ACKNOWLEDGMENTS

Many thanks to Dr. Sunil Kumar Koppurapu, my supervisor, for his help in shaping this paper.

## 6. REFERENCES

- [1] R. Sutcliffe, T. Crawford, C. Fox, D. L. Root and E. Hovy. 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] Cuthbert, M. S., & Ariza, C. 2010. music21: A toolkit for computer-aided musicology and symbolic music data. In *ISMIR 2010*, Utrecht, Netherlands, August 9-13 2010 (637-642).
- [3] Allam, A. M. N., & Haggag, M. H. 2012. The Question Answering Systems: A Survey. In *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3).
- [4] Gabriel, J. 2013. Large data sets & recommender systems: A feasible approach to learning music. In *Proceedings of the Sound and Music Computing Conference 2013, SMC 2013*, Stockholm, Sweden, p.701-706 .
- [5] Downie, J. S. 1999. Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text (Doctoral dissertation, The University of Western Ontario).
- [6] Viro, V. 2011. Peachnote: Music Score Search and Analysis Platform. In *ISMIR 2011*, Miami, Florida (USA) October 24-28, 2011 (359-362).
- [7] Ganseman, J., Scheunders, P., & D'haes, W. 2008. Using XQuery on MusicXML Databases for Musicological Analysis. In *ISMIR 2008*, Philadelphia, Pennsylvania (USA) September 14-18, 2008 433-438).