# UNLP at the C@merata Task: Question Answering on Musical Scores *ACM* [*]

Kartik Asooja
Insight Centre for Data Analytics
NUI Galway, Ireland
kartik.asooja@insight-centre.org

Sindhu Kiranmai Ernala
Center for Exact Humanities
IIIT Hyderabad, India
sindhukiranmai.ernala@students.iiit.ac.in

Paul Buitelaar
Insight Centre for Data Analytics
NUI Galway, Ireland
paul.buitelaar@insight-centre.org

## ABSTRACT

This paper presents a description of our submission to the C@merata task in MediaEval 2014. Our system accepts a query, recognizes the musical entities appearing in the query, and then searches them in the corresponding Music XML file. We submitted two runs for the task. First approach takes a union of the measures retrieved for each musical entity, while the second approach takes their intersection to answer the query. The recognition of the musical entities is based on regular expressions.

## 1. INTRODUCTION

This work explains the system submitted by us in the C@merata task at MediaEval 2014. The task targets natural language question answering on music scores. We were provided with a set of question types, and the data over which the search was required to be performed [2].

The questions in the task consist of short noun phrases in English referring to musical features in the music scores, for instance, "F# followed two crotchets later by a G". Every question refers to a single natural noun phrase using English or American musical terminology. The musical scores are provided in MusicXML, which is a standard open format for exchanging digital sheet music[1]. It allows the music scores to be machine readable. The music repertoire consists of Western Classical works from the Renaissance and Baroque periods by composers such as Dowland, Bach, Handel, Scarlatti etc. The answers comprise of zero or more passages from the music score, which contain the musical features mentioned in the query string. Basically, the answer points to the location(s) in the score of the requested musical feature. The answer passage consists of a start point and an end point in the music score associated with the query. A passage comprises of a start/end time signature, start/end divisions value, and start/end beat. The task provides two datasets, one for training and development while the other for testing. The training dataset consists of only 36 natural language queries, and the test set contains 200 questions.

## 2. APPROACH

There can be different types of musical features mentioned in the query such as note, melodic phrase etc. These different musical features can be referred as musical entities or can be defined with the help of such entities. So, we identify some of the basic entities from the natural language text, and perform the location search in the music score for the answer passages. In the current implementation, we aim to recognize basic musical entities only. For the complex ones requiring some combinations according to particular relations between the entities, we just take the union or intersection of the answer measures retrieved separately for different entities appearing in the query. Thus our approach consists of the following two main steps: Identification of Musical entities in the query, and retrieval of the relevant music passage from the provided Music XML file.

## 2.1 Identification of Musical Entities

We use regular expressions and created dictionaries to recognize musical entities in the query strings. The target entity types are:

### 2.1.1 *Notes*

A note defines a particular pitch, duration or dynamic, such as C, crotchet C, quarter note C in the right hand, semibreve C, whole note C, semiquaver F#, sixteenth note F#. The note recognizer comprises of three separate musical entity recognizers: duration, pitch and staff. We first recognize all the pitches appearing in the query string, and separately identify all the durations and staves. To identify the corresponding duration/staff for a pitch, we measure the string distance between all the pitches and duration/staff. The duration/staff, which occurs within a threshold distance from a pitch, is paired with that pitch in order to form the note. The pitches and durations are identified using regular expressions as shown in Table 1. We check the presence of the

[*]Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Conference'10, Month 1-2, 2010, City, State, Country.

**Table 1: Question Examples**

| Question Type | Count | Example Questions |
|---|---|---|
| Pitch | 30 | G5, E, B5, E#4 |
| Length | 30 | quarter note, dotted half note |
| Pitch & Length | 30 | crotchet F#, minim C sharp |
| Perf_spec | 10 | fermata A natural, staccato B flat |
| Stave_spec | 20 | Alto C#, G# in the Bass |
| Word_spec | 5 | minim on the word, G on the word |
| Followed_by | 30 | quaver C# followed by crotchet B, B followed by G |
| Melodic_interval | 19 | falling fourth, melodic fourth |
| Harmonic_interval | 11 | harmonic major tenth, harmonic minor sixth |
| Cadence_spec | 5 | perfect cadence |
| Triad_spec | 5 | tonic triad, triad in first inversion |
| Texture_spec | 5 | monophony, melody with accompaniment |
| Overall | 200 | |

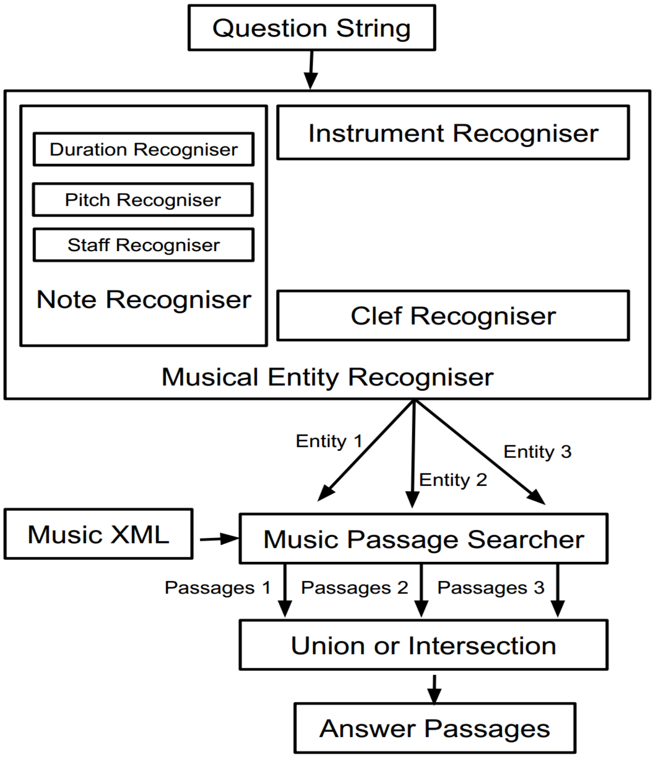

**Figure 1: Approach**

**Table 2: Question Examples**

<table>
<tr><th>Type</th><th>Regular Expression (Case Insensitive)</th><th>Examples</th></tr>
<tr><td>Pitch</td><td>(A|B|C|D|E|F|G|rest)\s*(#|b?)(\d)?\s*(sharp|natural|flat)</td><td>G5, E, B5, E#4</td></tr>
<tr><td>Duration</td><td>(dotted)?\s*(maxima|octuple\s*whole|longa|quadruple\s*whole|breve|double\s*whole|semibreve|whole|minim|half|crotchet|quarter|quaver|eighth|semiquaver|sixteenth|demisemiquaver|thirty-second|hemidemisemiquaver|sixty-fourth|semihemidemisemiquaver|hundred\s*twenty-eighth|demisemihemidemisemiquaver|two\s*hundred\s*fifty-sixth)\s*(note)?</td><td>crotchet, semiquaver, dotted minim</td></tr>
</table>

**Table 3: Results for different types**

| Question Type | Beat Precision Run 1\|2 | Beat Recall Run 1\|2 | Measure Precision Run 1\|2 | Measure Recall Run 1\|2 |
|---|---|---|---|---|
| Pitch | 0.422\|0.422 | 0.789\|0.789 | 0.478\|0.478 | 0.894\|0.894 |
| Length | 0.636\|0.636 | 0.797\|0.797 | 0.791\|0.791 | 0.991\|0.991 |
| Pitch & Length | 0.460\|0.460 | 0.696\|0.696 | 0.582\|0.582 | 0.880\|0.880 |
| Perf_spec | 0.0452\|0.0452 | 0.586\|0.586 | 0.0532\|0.0532 | 0.690\|0.690 |
| Stave_spec | 0.173\|0.173 | 0.440\|0.373 | 0.230\|0.241 | 0.587\|0.52 |
| Word_spec | 0.0667\|0.0667 | 0.833\|0.833 | 0.0667\|0.0667 | 0.833\|0.833 |
| Followed_by | 0.0\|0.0 | 0.0\|0.0 | 0.0252\|0.26 | 0.695\|0.633 |
| Melodic_interval | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 |
| Harmonic_interval | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 |
| Cadence_spec | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 |
| Triad_spec | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 |
| Texture_spec | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 | 0.0\|0.0 |
| Overall | 0.113\|0.29 | 0.516\|0.512 | 0.155\|0.393 | 0.703\|0.692 |

strings 'right hand' and 'left hand' in the query, in order to identify the staff value.

*Duration*
It defines the playing time of the pitch. In natural language, it can be defined as quarter, semibreve, whole etc. We simply write a regular expression covering the extensive vocabulary defining the duration in both English and American music terminology.

*Pitch*
It is a perceptual property that allows the ordering of sounds on a frequency-related scale. Some examples of writing pitches in natural language are: D sharp, A flat etc. We form a regular expression to identify the pitches in a query string.

*Staff*
To identify the staves mentioned in a string, we simply find the occurrences of "right hand" and "left hand" strings in it. The above three musical entities duration, pitch and staff collectively form the note entity.

### 2.1.2 Instruments

We manually created a dictionary of instruments using the training and test data, in order to match it against the words in the query string. The dictionary includes the words like viola, piano, alto, violoncello, soprano, tenor, bass, violin, guitar, sopran, alt, violin, voice, harpsichord etc.

### 2.1.3 Clef

To identify the Clef, we just check the presence of strings like bass clef, F-clef, treble clef and G-clef in the query. The implementation including the regular expressions and the dictionaries used can be found at the publicly available code repository at GitHub.

## 2.2 Music Passage Retrieval

The identified musical entities in the query are searched in the music score associated with the question. The identification of the musical entities remains same in both the submitted runs. They just vary on the basis of the following two approaches for music passage retrieval:

- Union of the musical measures that contain the target musical entities is used to create the answer passages.

- An intersection of the musical measures that contain the target musical entities is used in the answer passages.

## 2.3 Evaluation

The system performance is measured for each question type, and an overall weighted average for all the questions is also calculated. Table **??** shows the results obtained by our two runs. As discussed in the approach section, the current implementation only recognizes a few types of musical entities, which constraints the question types to be answered. The results clearly show that the system could not answer many question types like melodic, harmonic, cadence etc. It is because detection of such musical features was not implemented in the current system.

### 2.3.1 Results and Discussion

In the current version, our system only uses string and regular expression matching for the identification of musical elements, while string distance is used to identify the relations between the elements, if required. However, there is a scope of deep syntactic and lexical analysis of the query string to identify the relations between the entities. We also found a small bug in our system related to the 'natural' appearing in a query string, which lead to some wrong answers, because of incorrect octave calculation.

## 2.4 Conclusion

The presented approach presents an initial implementation of the natural language question answering to musical scores. The pipeline is based upon identifying the different types of musical entities in the query string, and their relations. The results can be greatly improved by incorporating Natural Language Processing based methods for better understanding the query and identifying the relations between the entities. We consider it as a future direction to explore.

## 2.5 Acknowledgement

last We are very grateful to Mr. Robert Solyom from Music Academy, Galway for helpful suggestions and references.

## References

[1] M. Good and L. C. Recordare. Lessons from the adoption of musicxml as an interchange standard. 2006.

[2] R. Sutcliffe, T. Crawford, C. Fox, D. Root, and E. Hovy. Shared evaluation of natural language queries against classical music scores: A full description of the c@merata 2014 task. *Proceedings of the C@merata Task at MediaEval 2014*.