# TNKG at the MediaEval 2015 C@merata Task

Nikhil Kini
Mumbai
India
nikhil.n.kini@gmail.com

## ABSTRACT

This paper explores a method to answer natural language musicological queries against electronic music scores, with the aim of developing a complete grammar specification in the domain. Our system takes a three step approach to finding the answers to the queries - first, it replaces the musical features in the question with our own tags in a key value form, and also replaces words in the question with equivalents from a synonym list. This is to normalize the natural language and minimize the search space. Second, based on the word sequence in the normalized question, an inference on the type of question is made using handwritten rules. Finally, we have functions that search through the MusicXML for the answer based on the question type.

## 1. INTRODUCTION

The C@merata task [1,2] aims at interpreting natural language queries on electronic music scores to aid musicologists in locating occurrences of musical features of interest at precise points in Western classical sheet music. The task is of special interest to students and professors of Music theory, to the music retrieval community and Musicologists studying similarity in the works of composers.

This paper continues the work presented in [3], approaching the problem of natural language understanding in the following manner: a) convert the NL query into tokens, b) parse the tokenized query to identify the type of query, c) use search functions to search the MusicXML for answers based on the type of query.

This is the second year of the C@merata task as well as of our participation. We found the questions this year were more complex than last year, at least linguistically, which made adjustments and improvements to the system necessary. There were on average approximately 7 words per question, up from approximately 3 words per question in 2014. In particular, the vocabulary of questions increased to 253 unique words from 116. Consequently, we updated our synonyms list, token-searching regular expressions and question inference regular expressions to make allowance for this increase in complexity.

We show how we have extended the tokens in this paper, and we also briefly describe work in progress to create a grammar that is capable of generating the language given by the set of available questions.

## 2. APPROACH

Our approach remains the same as last year's - the outline is shown in Figure 1. Musical features in the question are identified in the Token identification module, and also a list of words to be replaced is held in the synonyms list. The tokenized sentence is analyzed by the question type inference module based on the question rules to infer what kind of a question the query might be. Next, a Python function corresponding to the inferred question type searches through the MusicXML to arrive at an answer.

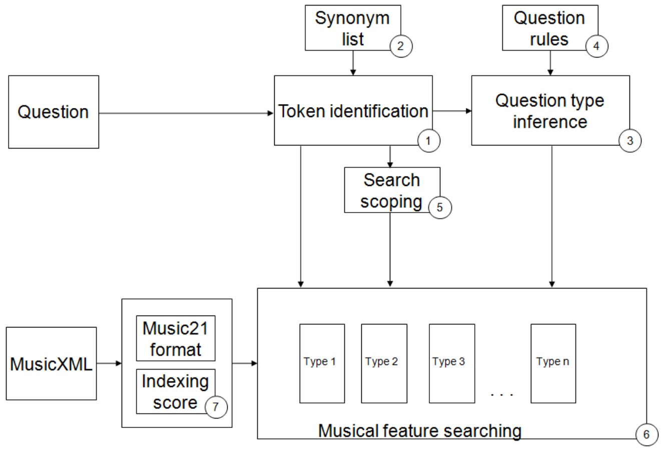


**Figure 1**

A brief explanation of the modules and improvements in them follows.

## 2.1 Tokenizing and Synonyms

In the tokenizing step, we mark musically relevant features with 3 or 4 letter markers that we call the token class. E.g., quarter note is a duration, and is marked DUR, and octave is an interval, and is marked INT. 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)". Changes to the module this year included 1) new musical features were added, culled from the test set in 2014, such as the double-sharp 'x' being added to sharps and flats; 2) intervals like seventeenth and nineteenth were added; 3) SCP (scope) was added as a token to explicitly mark the scope of the search.

## 2.2 Question rules and inferring the question

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. While we introduced no new question types, we worked on expanding the implementation of the "Combination of the above" type.

We made an attempt at grammar inference of the natural language queries. Grammar inference (or grammar induction) is the process of learning a formal grammar by observing the samples of language that are presented [4]. This is our long term goal with the task - to parse the question to arrive at a question type rather than through regular expression rules, and to be able to dynamically extend the queries that are understood by the system, thereby making it a self-learning system. Here we briefly discuss


*MediaEval 2015 Workshop*, September 14-15, 2015, Wurzen, Germany.

the present attempts and future work we intend to do with grammar inference.

Grammar inference (or grammar induction) is the process of learning a formal grammar by observing the samples of language that are presented [4]. Consider a short demonstrative example where the language consists only of durations and pitch values, and queries such as:

Q1: Quaver
Q2: Whole note C#

A grammar for such a language is:

---

| | | |
|---|---|---|
| S | → | NOTE |
| NOTE | → | P D P |
| P | → | PTCH_CLS OCTV ACCDNTL \| ∊ |
| D | → | breve \| semibreve \| quaver \| whole \| half \| quarter … \| ∊ |
| PTCH_CLS | → | A \| B \| C \| D \| E \| F \| G |
| OCTV | → | 0 \| 1 \| 2 \| 3 \| 4 \| 5 \| 6 \| 7 \| 8 \| 9 … \| ∊ |
| ACCDNTL | → | # \| ## \| b \| bb \| x \| flat \| sharp \| double sharp \| ∊ |

---

A total of 419 questions were available, sourced from examples in specifications, training sets, sample questions, and the 2014 Gold standard test set. From the 419 questions, we have a vocabulary of 253 words. These are a subset of the terminal symbols in the grammar. The entire set would contain all synonyms and musical terms that do not appear in our sample questions. It is important to note that there may be out of vocabulary words from sources such as lyrics for which it will be difficult to account in the grammar. We use the UNK token to mark out of vocabulary words, and as far as possible, include it in the production rules. For example, minim on the word "Der" can be represented by the production rule:

---

| | | |
|---|---|---|
| NOTE_ON_LYRICS | → | NOTE SCOPE LYR |
| NOTE | → | P D P (same as above) |
| SCOPE | → | on the word \| after \| over the lyric |
| LYR | → | UNK |

---

In our work, experiments were carried out to see if production rules could be learnt using the questions available to us. Unfortunately, this did not succeed. At the time of writing this paper, we know no software that can automatically infer grammar from positive examples. We tried two softwares, GI-Toolbox (https://code.google.com/p/gitoolbox/) and Sequitur (http://www.sequitur.info/). However, we could not adapt them to the alphabet of Musicological terms in time. So we used a manually created grammar. Our objective is to parse the tokenized representation of each query into a question type, and each question type can be mapped to a retrieval function that can provide answers. The parameters of the retrieval functions are the specific values for the non-terminal symbols in the parse tree.

After writing a formal grammar, we can use a parser to parse sentences in the language with the use of ANTLR (http://www.antlr.org/). The parser is then used on the input natural language queries to obtain a parse tree. The root of the parse tree is the question type.

### 2.3 Search scope

Search scope is important to increase the precision of the system by helping to retrieve not just the requested musical feature, but to do so within the specified context in the query. This year's approach to setting questions also made this scope explicit in the form of specifying the basic question types and modification of these questions in six different ways - performance, instrument, clef, time, measure/bar and key. We updated our system to include the SCP token so that scope could be recognized and search could be restricted to only the part of music that satisfies the scope.

### 2.4 Searching for the answer

Once the question type is inferred, the last step is searching the MusicXML score for the identified token/token combination. We added the harmonic analysis parts for the score which was missing last year. Updates to the music21 [5] library enabled us to find "verticalities" in the score - information about the active notes at a particular time interval slice. This was analyzed to search for requested harmonic features in the score.

### 2.5 Score index

This is a list of all the notes in the score, stored with metadata about each note - implemented as a reverse dictionary for quick retrieval - and there was no difference in this from last year.

## 3. RESULTS AND DISCUSSION

We did not expect to perform very well on the retrieval task since there was insufficient time to complete the search functions within our system by the deadline. Results from the submitted run show that though we have decent beat recall and and measure recall, the precision is poor. The beat precision and measure precision values for the entire run are 0.0605 and 0.0727 respectively, and the beat recall and measure recall are 0.488, 0.586. The best recall is achieved for the 1_melod type where a beat recall of 0.865 and measure recall of 0.908 were seen. Forty questions were accurately answered, both beat and measure recall and precision being 1. Twenty-nine of these were of the simple 1_melod type. However, questions like "dotted quarter note Eb followed by sixteenth note F", "G#3 in the "III" part" and "dotted quarter note in 3/8 time" were also answered exactly. Further, 40 more questions were answered with a beat recall and measure recall of 1, but precision over a range of 0.004 to 0.9. Most of these were 1_melod modified by perf, instr, clef or time. Clearly, the system in its current iteration is decent at recognizing notes, but work needs to be done on other aspects of musical concept retrieval.

## 4. CONCLUSION

Although our results were not improved over last year, we believe progress has been made towards our goal of reaching a formal grammar specification of natural language queries on Western sheet music. We will continue with the implementation of the pending retrieval functions and the parser for the grammar, along with improvements to the grammar, and are confident of achieving strong results next time.

## 5. REFERENCES


[1] Sutcliffe, R. F. E., Fox, C., Root, D. L., Hovy, E. and Lewis, R. 2015. The C@merata Task at MediaEval 2015: Natural language queries on classical music scores. In MediaEval 2015 Workshop, Wurzen, Germany, September 14-15, 2015. http://ceur-ws.org.

[2] 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. http://ceur-ws.org/Vol-1263/.

[3] Kini, N. (2014). TCSL at the MediaEval 2014 C@ merata Task. In MediaEval 2014 Workshop, Barcelona, Spain, October 16-17 2014. http://ceur-ws.org/Vol-1263/.

[4] De la Higuera, C. (2010). Grammatical inference: learning automata and grammars. Cambridge University Press.

[5] 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).