

# CLAS at the MediaEval 2015 C@merata Task

Stephen Wan  
Language & Social Computing Team  
CSIRO  
Sydney, Australia  
stephen.wan@csiro.au

## ABSTRACT

The CLAS 2015 system treats the C@merata task as a Q&A problem specified with a controlled language. In this year's system, we added a context-free grammar for the music controlled language using the Natural Language ToolKit. Crucially this provides an in-built feature unification mechanism allowing us to replace the ad-hoc unification component in the 2014 system. The CLAS 2015 system with this modification finished first in the C@merata shared task. In this paper, we describe the approach behind our participation in the shared task and discuss arguments for and against using a feature-based context-free grammar to parse queries.

## 1. INTRODUCTION

The C@merata task [1] provides an opportunity to investigate natural language queries to structured data, in this case, music data. This data, which is akin to time-series data, is composed of sequenced events, each with associated metadata.

In contrast to the 2014 task [2], this year's shared task included complex queries with constraints to restrict candidate answers. For example, the query "ten staccato quarter notes in the Violoncello in measures 1-60 followed by two staccato quarter notes in the Violin 1" requires finding two answer sequences that are juxtaposed together. Furthermore, the answer sequences occur in different musical parts played by the violoncello and the first violin.

The CLAS 2015 system, like its 2014 predecessor [3], is based on the general notion of unification between the lexico-semantic features of the query and the metadata for each musical event. In brief, the system interprets a natural language query, converting the query to a conceptual representation. This is in turn processed to form a query representation, defining the type of answer required. Feature unification is generally used to find a subset of the data that serves as a candidate answer.

In the CLAS 2014 system, the components to detect linguistic features and unify these with metadata was purpose-built for the C@merata 2014 task. In particular, our system did not heavily rely on phrase structure in the query when extracting linguistic features to match against --- aside from specific nouns indicating the beginning of a new noun phrases subsequence, no other phrase structure was inferred.

However, the inclusion of more complex queries provided some cases where a more complex syntactic phrase structure is required to adequately represent the meaning of the query. For example, numbers can be used to refer to a specific bar (for example, "a note in bar 4"), to specify a range of bar indices (for example, "a note in bars 1 to 4") or to indicate cardinality (for example, "4 crotchets"). Syntactic structure can help in these cases to interpret the query correctly.

Consequently, this year, the CLAS system uses the natural language feature-based parsing facilities in the Python modules distributed as part of the Natural Language ToolKit (NLTK) [4] and a Context-Free Grammar (CFG) defined by the author. One side-effect of this approach is that the feature unification facility of the parser can be used to match against feature structures based on the music events.

The CLAS 2015 system achieved 0.60 precision and 0.63 recall when specifying answers at the granularity of "beats" in a bar (more accurately, a subdivision of a beat, as specified by the question). When examining accuracy at the granularity of bars, our system achieved a 0.64 precision and 0.67 recall.

In the remainder of this paper, Section 2 describes the overall system with an emphasis on how we employ NLTK and how we designed the feature-based CFG. Section 3 presents an overview of the system's performance in the C@merata 2015 tasks with a preliminary discussion of how our approach fared with different query types. We end with some final comments in Section 4.

## 2. APPROACH

### 2.1 Strategy for participation

Our general approach to participation for the 2015 entry was to port the domain-specific rules mapping from a query to a conceptual representation in the CLAS 2014 system to feature-based CFG for use with NLTK's parsing tools. In the advent of an out-of-vocabulary error or an empty parse, the system reverted to the 2014 system.

This grammar was then extended to cover the complex queries of the 2015 shared task. We compiled a list of the new queries from the documentation of the shared task. Grammar development was done by checking to see that there was an intuitive parse and that this led to candidate answer. The correctness of the answer was vetted manually.

This year, we had the benefit of a training data set. We set up a simple evaluation framework to gauge if changes to the CFG corresponded to overall improvement. As we did not have the evaluation code to measure precision and recall (and we did not have sufficient time to implement our own), we used the diff tool (with the `-w` option to ignore whitespace) to compare between the 2014 gold standard and our system results. The number of different lines was used as a rough measure of performance: fewer lines was taken as a indicating a grammar with better coverage.

### 2.2 Designing a Context-Free Grammar

In general, the CLAS 2015 grammar models the query as a nested sequence of musical noun phrases. These phrases are based predominantly on the basic noun phrases that were handled in the 2014 CLAS system but extended to include new aspects for 2015 such as chords in a specific key, solfège nomenclature for notes, and references to scales. No morphological analysis was performed and plurals were hardcoded into the lexicon. A nested

semantic feature structure was propagated to the root to allow for matching against the data.

Question Type	Example
l_melody	<b>dotted minim F#4</b>
n_melody	<b>five</b> note melody in bars 1-10
l_harm	<b>chord D2 E5 G5</b> in bars 54-58
texture	<b>monophonic passage</b>
follow	A minim <b>followed by</b> a quaver
synch	chord C4 E4 <b>against</b> a C5
perf	<b>sforzando</b> F2
instr	harmonic second <b>in the Violin 2</b>
clef	four Gs <b>in the treble clef</b>
time	F sharp <b>in 6/8 time</b> in bars 1-20
key	sixteenth note G <b>in G minor</b>

**Table 1. Query types and examples.**

We used domain-specific inferences to handle the queries. These depended, in part, on where preposition phrases are attached. For example, a preposition at the root of the parse was used for restrictions to bars and parts. Prepositions attached to sequence-based phrase constituents (typically the last element in the sequence) were used to represent metadata constraints that should be inherited by all elements in the sequence (for example, “C, D, E in crotchets”). Finally, a prepositional phrase in the noun phrase for the musical event itself was used to qualify the metadata (for example, “a chord in C”).

Referring expressions proved to be a minor complication as we did not want to hardcode every enumerated object, such as a number, in the lexicon. We replaced numbers with a placeholder token “\_NUM\_” during the parsing process. The actual numeric value was then heuristically reinserted into the parse structure. The same mechanism was used for lyrics and enumerated part names like “Violin II”.

When phrases like “followed by” were detected in the query, we split the query at that point to form two component queries. Each query was then treated independently and all candidates that were adjacent with respect to its time (bar and beat indices) were considered an answer.

### 3. RESULTS AND DISCUSSION

The evaluation was divided into a number of different query types. Examples of these are presented in Table 1 with the key elements in bold. We present the official evaluation results in Table 2. Results are reported for both beat and bar granularities.

In 2014, the worst category for our system was harmonic intervals and so we focused on this category, adding domain-specific inferences about notes, chords and scales to our handling of harmonic intervals. We are pleased to see that this led to good performance for this category.

Our approach to splitting a query into component parts that are juxtaposed led to good precision for the “synch” category but leaves room for improvement in terms of recall. Our inferences mechanism based on the attachment points of prepositional phrases led to the reasonable performance for the “instr”, “clef” and “time” query categories.

We ignored cadences and texture in this year’s effort and our performance suffers correspondingly for these query types. Due

to time restrictions, we were unable to add domain-specific knowledge to handle musical references such “Alberti bass”, “arpeggios”, and “descending scale”. We were also only partially able to handle the restriction “melody” as in “melody C, D, E in the violin”, as this requires an analysis of texture.

Question Type	Beat Prec.	Beat Recall	Bar Prec.	Bar Recall
l_melody	0.655	0.812	0.687	0.852
n_melody	0.716	0.52	0.77	0.559
l_harm	0.66	0.62	0.702	0.66
texture	0	0	0	0
follow	0.312	0.484	0.323	0.5
synch	0.818	0.25	1	0.306
perf	0.955	0.467	0.955	0.467
instr	0.677	0.708	0.72	0.753
clef	0.415	0.519	0.431	0.538
time	0.679	0.905	0.75	1
key	1	0.625	1	0.625

**Table 2. Evaluation results for different query types.**

Evaluation performance aside, it is worth reflecting on the strengths and weaknesses of our approach. Our NLTK-based system was notably slower on the 2014 training data set compared to our 2014 version. It is difficult to say which system is easier to maintain and develop. Intuitively, we believe the CFG may be easier to maintain given the ease of porting the 2014 resources to a CFG and the ability to write domain-specific rules based on phrase structure.

One limitation is that we are only able to handle queries licensed by the grammar, meaning we are unable to handle ungrammatical query. This is potentially too prescriptive, particularly if this were to be a real application. Our CFG deliberately allows metadata for notes to be accepted in any order (for example, “minim dotted # C4”) but this is the extent to which we accept an ungrammatical query. Finally, we found that feature unification as a paradigm for matching against metadata breaks down at times. The simplest case is that of intervals, the metadata value of the note name for the second note depends on the context of another note. For example, a “perfect fifth” is not always a “C, G” pattern. Enumerating all fifths seems inelegant. For these cases, other answering mechanisms are needed.

### 4. CONCLUSION

The CLAS 2015 system treats the C@merata task as a Q&A problem using a controlled language. We use a feature-based context-free grammar to define a controlled language for the music domain and parse queries using the Natural Language ToolKit. The CLAS 2015 system with this modification finished first in the C@merata shared task.

### 5. ACKNOWLEDGEMENTS

We would like to thank the organisers of C@merata for such an engaging, enlightening, and yet entertaining shared task.

## 6. 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] Stephen Wan. 2014 The CLAS System at the MediaEval 2014 C@merata Task. In the Working Notes Proceedings of MediaEval 2014 Workshop. Barcelona, Catalunya, Spain, October 16-17, 2014, CEUR-WS.org
- [4] Bird, Steven, Edward Loper and Ewan Klein. 2009, Natural Language Processing with Python. O'Reilly Media Inc.