Lang Resources & Evaluation
https://doi.org/10.1007/s10579-018-9422-2

ORIGINAL PAPER

Searching for musical features using natural language queries: the C@merata evaluations at MediaEval

Richard Sutcliffe[1] · **Eduard Hovy**[2] · **Tom Collins**[3] · **Stephen Wan**[4] · **Tim Crawford**[5] · **Deane L. Root**[6]

Abstract Musicological texts about classical music frequently include detailed technical discussions concerning the works being analysed. These references can be specific (e.g. C sharp in the treble clef) or general (fugal passage, Thor's Hammer). Experts can usually identify the features in question in music scores but a means of performing this task automatically could be very useful for experts and beginners alike. Following work on textual question answering over many years as co-organisers of the QA tasks at the Cross Language Evaluation Forum, we decided in 2013 to propose a new type of task where the input would be a natural language

✉ Richard Sutcliffe
rsutcl@essex.ac.uk

Eduard Hovy
hovy@cmu.edu

Tom Collins
collinte@lafayette.edu

Stephen Wan
stephen.wan@data61.csiro.au

Tim Crawford
t.crawford@gold.ac.uk

Deane L. Root
dlr@pitt.edu

[1] School of CSEE, University of Essex, Colchester, UK

[2] Language Technologies Institute, Carnegie-Mellon University, Pittsburgh, PA, USA

[3] Department of Computer Science, Lafayette College, Easton, PA, USA

[4] CSIRO, Epping, NSW, Australia

[5] Department of Computing, Goldsmiths, University of London, London, UK

[6] Department of Music, University of Pittsburgh, Pittsburgh, PA, USA

Published online: 12 August 2018

phrase, together with a music score in MusicXML, and the required output would be one or more matching passages in the score. We report here on 3 years of the C@merata task at MediaEval. We describe the design of the task, the evaluation methods we devised for it, the approaches adopted by participant systems and the results obtained. Finally, we assess the progress which has been made in aligning natural language text with music and map out the main steps for the future. The novel aspects of this work are: (1) the task itself, linking musical references to actual music scores, (2) the evaluation methods we devised, based on modified versions of precision and recall, applied to demarcated musical passages, and (3) the progress which has been made in analysing and interpreting detailed technical references to music within texts.

Keywords Question answering · Natural language processing · Music information retrieval · Musicological analysis · MusicXML · Evaluation

1 Introduction

In Western classical music, a work is normally written down by the composer in music notation. Trained musicians can read a score in this notation and hence perform the work. Moreover, musicologists and other experts can also read this notation for the purposes of study or analysis. The results of such study are usually then written in a natural language so that others can benefit from the insights so gained.

The modern system of musical notation involving staff lines and clefs is often attributed to Guido d'Arezzo, but according to Read (1978), it was not until well into the seventeenth century that the system started to reach a standard form. Since that time, the process of refinement has continued in order to match the needs of both composers and performers. In parallel with this, the means of referring to features of a score using natural language has also reached a comparable level of complexity and sophistication. The aim of the Cl@ssical Music Extraction of Relevant Aspects by Text Analysis (C@merata) evaluations is to explore further the exact nature of this connection between language and music and to encourage the development of mechanisms for linking them together. We thus combine Natural Language Processing (NLP) and Music Information Retrieval (MIR) in an unusual way.

The first C@merata campaign was announced in early 2014 and ran until September of that year (Sutcliffe et al. 2014a, b). There were twenty classical music scores with ten questions being posed against each. A question took the form of a short noun phrase referring to some feature in the corresponding score, e.g. 'C sharp in the treble clef'. The required answer was a set of one or more passages—each a demarcated portion of the musical score which exactly encodes the required feature, in this case the C sharp.

In 2015 and 2016 there were two further editions of C@merata (Sutcliffe et al. 2015b, c, 2016). The overall task and organisation remained the same, but questions became steadily more complex.

In this article, we first describe related previous work which combines natural language processing with Music Information Retrieval (MIR). We then explain the task in more detail, including the design of the questions, the passage concept, the means devised for automatic evaluation and the choice of music score format. Next, we turn to the detailed design of the data sets for each year of the campaign. This includes the choice of music scores, the design of the questions and the determination of the correct answers to be used for evaluation (i.e. the Gold Standard). We then outline the actual campaigns for each year, including details of the participants, the mechanisms used in their systems and the results obtained. Finally, we analyse the results for the 3 years, summarise what the three campaigns have achieved and assess what should be done next.

2 Previous work

The origins of C@merata lie in two main branches of work: Question Answering (QA) and the analysis of popular song lyrics. Furthermore, we also discuss more recent Information Extraction work on documents about music.

QA evaluations have proved a popular way of encouraging rapid developments in natural language processing and information extraction. Between 2004 and 2010 we were co-organisers of the QA task at the Cross-Language Evaluation Forum (CLEF).[1] The input was a natural language query together with a large document collection, and the required answer was a Named Entity (Nadeau and Sekine 2007) or short text. There were 200 questions and registered participants could download them on a specific date and upload their results for evaluation by the organisers. Evaluation was carried out manually, following the Text REtrieval Conference (TREC)[2] model (Voorhees 2002). Question Answering for Machine Reading Evaluation (QA4MRE) was its successor at CLEF between 2010 and 2013 (Peñas et al. 2013; Sutcliffe et al. 2013). The input was a more complex query together with a document collection, and the required answer was in multiple-choice form (five possible answers) which enabled automatic evaluation. For 2011, questions were organised into three topics, AIDS, Climate Change and Music and Society. There were four documents on each topic and ten MCQ questions for each document. In 2012 the topic of Alzheimer's was added with once again four documents per topic and ten questions each. Finally, in 2013, there were the same four topics with four documents on each, but fifteen questions per document instead of ten. Table 1 shows examples from the Music and Society topic, taken from each year of the QA4MRE task.

In 2011, all documents were transcriptions of talks at the Technology, Entertainment, Design (TED) conferences.[3] For Music and Society, the four documents were Adam Sadowsky: 'Adam Sadowsky engineers a viral music video', Ben Cameron: 'The true power of the performing arts', David Byrne: 'How

[1] http://www.clef-initiative.eu/.

[2] http://trec.nist.gov/.

[3] http://www.ted.com.

Table 1 Sample questions used in the QA4MRE evaluations at CLEF, 2011–2013. The three columns are (1) the type, (2) the question text, (3) the MCQ answers (correct one is starred)

2011 QA4MRE questions		
CAUSE	Why can Bach change key without risking dissonances?	the room had an organ / there was no rhythm / the room was smaller * / the room was larger / the music was perfect
FACTOID	What is the full name of El Sistema?	The National System of Youth and Children's Orchestras and Choirs * / Venezuela's Youth Symphony Orchestra / Simon Bolivar Youth Orchestra of Venezuela / Arnold Toynbee / Mother Teresa of Calcutta
METHOD	In the "10 commandments of music video", how was the action of the machine to follow the feeling of the song?	when the song became slower, the machine was to become faster / when the action became grander, the song was to become slower / when the video became more popular, the song was expected to sell more copies / when the song became more emotional, the machine was to work on a larger scale * / when the dancing was on treadmills, the machine was to work faster
PURPOSE	Why did the Bayreuth Festspielhaus have a large orchestra pit?	to eat, drink and yell out / to be more intricate / to help Mozart / to suggest an encore / to accommodate low-end instruments *
WHICH IS TRUE	What was made possible in music when audiences were forced to be quiet?	Bob Dylan's last record / gossiping and shouting at all times / articles in the New Yorker by Alex Ross / extreme dynamics * / the music of Scott Joplin
2012 QA4MRE questions		
CAUSE	Why was the fourth volume of the History of Music [by Charles Burney] not satisfactory?	German music was not covered at all / There was too much material on Handel and Bach / There was insufficient material on Handel and Bach * / Greek music was not discussed / Handel and Bach were not mentioned
FACTOID	What invention is attributed to Giuseppe Torelli?	the church / the orchestra / the concerto * / the band / the leader
METHOD	How did Lulli conduct?	He struck his foot / He lived in Paris / He was avaricious / He used a cane * / He had great ability
PURPOSE	Why, when playing at the house of Cardinal Ottoboni, did Corelli stop playing in the middle of a solo?	Because Ottoboni was talking and not listening * / because Ottoboni liked music / because he wished to join the conversation / because he was in Rome / because he was a violinist

Table 1 continued

WHICH IS TRUE	What did Forkel think about Burney's History of music?	He criticised it * / He liked it / He praised it / He understood it / He misunderstood it
2013 QA4MRE questions		
CAUSE	Why did Schenker's mature theory only emerge after many years of struggle?	because musical analysis is very easy / because Schenker was lazy / because musical analysis is very difficult * / because Schenker was busy with other projects
FACTOID TIME	When did the author of "The Sandman" write a review of a Beethoven symphony?	1810 * / 20th century / 1957 / 1986
METHOD	How is analysis carried out?	determine the composer and date / determine the influences on a composition / determine the parts of a composition and links between them * / determine the links between the composition and others
PURPOSE	What is the prime concern of analysis?	to determine the structure and features * / to determine the length and era / to determine the structure and era / to determine the features and era
WHICH IS TRUE	How well-known are Cramer's late sonatas today?	the sonatas are universally known / the sonatas are very widely known / the sonatas are widely known / the sonatas are little known * / none of the above

architecture helped music evolve', and Jose Abreu: 'Jose Abreu on kids transformed by music'. In 2011, Music and Society documents were taken from public domain sources, namely Federal Reserve Bank of Boston, Gutenberg,[4] 1911 Encyclopedia[5] and Wikipedia.[6] Documents were 'Requiem for Classical Music', 'Famous Violinists of Today and Yesterday', 'Charles Burney' and 'Pop Music'. In 2012, Music & Society documents were all taken with permission from Grove's Dictionary of Music and Musicians (Grove Music Online),[7] namely 'Disciplines of Musicology—Analytic Traditions', 'Electronic Dance Music', 'Film Music—Hollywood' and 'Johann Baptist Cramer'.

The questions asked about music at QA4MRE were of several different kinds (Table 1). Cause questions sought the reason for something ('Why did Schenker's mature theory only emerge after many years of struggle?'). Factoids asked for information such as a person, place or time ('When did the author of "The Sandman" write a review of a Beethoven symphony?'). Method questions asked how something was done ('How is analysis carried out?'). Purpose questions asked why something was done ('What is the prime concern of analysis?'). Finally, Which-Is-True questions required the choice between several alternatives posed as direct statements ('How well-known are Cramer's late sonatas today?').

The QA4MRE work enabled us to gain some insight into the music domain from the perspective of music texts, viewed from an NLP perspective—our knowledge of such texts had previously been from an entirely musicological standpoint. We could see that there were numerous Named Entities, many of a very specialised kind, that certain syntactic constructions were used in preference to others, and that there were very detailed linguistic means of referring to musical events. This led us to propose the C@merata task and in addition to carry out further, more-detailed analyses of musicological texts (Sutcliffe et al. 2015a).

We now consider research by others which relates to C@merata. Previous work which combines NLP with MIR can be divided into various categories. Firstly there have been applications of Information Retrieval techniques such as the Vector Space Model (VSM) (Salton et al. 1975) to text relating to artists or songs. For example, Baumann (2003) collected documents about a popular artist and then computed a standard VSM vector for them. Information Retrieval distance measures could then be used to compare artist vectors and hence recommend similar artists to one known to be of interest to the user.

The main work combining MIR and NLP, however, has been to extract information from song lyrics for purposes such as recommendation or mood classification (Logan et al. 2004; Mahedero et al. 2005; Sterckx et al. 2014; Buffa and Cabrio 2016). Logan et al. (2004) used Probabilistic Latent Semantic Analysis to learn a set of topics from a large collection of song lyrics. Hence, the lyrics of songs by a particular artist were used to compute a vector representation of that artist which could subsequently determine similarity between artists.

[4] http://www.gutenberg.org/.

[5] http://en.wikisource.org/wiki/Wikisource:WikiProject_1911_Encyclop%C3%A6dia_Britannica.

[6] http://www.wikipedia.org/.

[7] http://www.oxfordmusiconline.com/grovemusic/.

Mahedero et al. (2005) carried out four types of analysis on song lyrics: language recognition, structure extraction, thematic categorisation and similarity searches. Their work followed Baumann (2003) and Logan et al. (2004) who used information from lyrics for music browsing. Language recognition used the method of Dunning (1993) based on letter n-grams. Structure extraction was performed by computing the similarity between paragraphs, detecting repeated paragraphs and unique ones, and finally applying rules of song composition (e.g. a song probably starts with an introduction) to posit the overall structure. This was done on songs in five different languages. Thematic categorisation was performed using Naive Bayes and working with five categories, namely Love, Violent, Protest, Christian and Drugs. Finally, song similarity was computed using cosine distance on standard Information Retrieval document vectors. The experiments did not use much NLP but they did show that a lot of interesting data could be extracted from song lyrics.

Sterckx et al. (2014) used Unlabelled and Labelled Latent Dirichlet allocation to create topic models for song lyrics. An analysis was then carried out to determine their usefulness for tasks such recommendation. Finally, concerning the use of song lyrics alone, Buffa and Cabrio (2016) describe an ongoing project aiming to analyse the words of a song in order to detect the structure, determine the time references and named entities used, and establish the overall topic or abstract themes.

Some of the work using data from song lyrics has also combined it with information obtained from audio and other sources (Wang et al. 2004; Whitman and Ellis 2004; Hu et al. 2009; McKay et al. 2010; Mihalcea and Strapparava 2012) or from symbolic music (Brochu and de Freitas 2003). Wang et al. (2004) aligned the text lyrics of a song with its audio recording, which also contained the song lyrics. Timing and rhythm data were extracted from the audio as were the portions of it where singing was taking place. Text processing on the lyrics identified verses by blank lines and estimated the length of each line based on a phonetic transcription. Text and audio were then aligned in a two-stage process: they first aligned per verse in the song, and then per line.

Whitman and Ellis (2004) analysed textual song reviews and combined this data with audio features. N-grams, adjective sets and noun phrases were extracted from the reviews. So-called 'Penny' features (i.e. ones derived from Mel-frequency Cepstral Coefficients at a 100 Hz sample rate) were extracted from the audio of the corresponding songs. The two were then linked using the Regularized Least-Squares Classification (RLSC) machine-learning method. The association was between the text and the whole song, and the aim was to generate review phrases automatically from the recording.

Hu et al. (2009) were concerned with mood classification of complete songs based on the combined use of lyric features and audio spectral features. McKay et al. (2010) investigated song genre classification and they compared the performance of low-level features extracted from lyrics (e.g. the words they contain, part-of-speech frequencies etc.) with other information including audio data, symbolic data and cultural information from the internet. Results were reported for both a five-genre taxonomy and a ten-genre taxonomy.

Mihalcea and Strapparava (2012) made a corpus of 100 popular songs such as Let it Be. For each song, they used crowdsourcing to annotate each text line with the

degree of 'raising' in the music corresponding to that line, and the degree of affect expressed, using six emotions proposed by Ekman (1993), namely Anger, Disgust, Fear, Joy and Surprise. Raising is the interval between the first note and the longest note in a phrase. Then, they attempted to predict the emotions in a line using the text only, the music only and the text and music together. The method used was linear regression, evaluated using Pearson correlation.

For the textual analysis, Mihalcea and Strapparava used unigrams (i.e. words in the text with low frequency words removed) and semantic classes for words taken from the Linguistic Enquiry and Word Count and WordNet Affect data sets. For training, each line is therefore represented by one or more unigram features and one or more semantic class features. For the musical analysis, they use the frequency of each note within the line, a note being chosen from twelve, i.e. ignoring octave pitch. The key of the entire song is also a feature. The textual features alone were better at predicting emotion than the musical features alone. However, using both feature sets produced the best results, the highest Pearson correlation being 0.67 for Anger.

This work is particularly interesting in relation to ours because it directly combines information from the text (words and classes) with information from the music score itself (the Raising, note frequency, the key of the song).

Brochu and de Freitas (2003) modelled both lyric text and symbolic music scores in the GUIDO[8] notation. The music was encoded in a Markov model which included intervals and note lengths. The associated text was modelled as a term-frequency vector (i.e. a Vector Space Model). Text could be the lyrics of a song or a text description of the music e.g. from the Internet. The parameters of the Markov model were estimated using Expectation Maximisation. The training data included monophonic songs with lyrics and some Bach inventions. Searching was at the level of whole songs or whole texts, so this work was a form of multimodal information retrieval. This work is unusual in using symbolic music scores; audio tracks are mainly favoured because of the commercial pressure to retrieve or recommend popular music songs for the mainstream Internet user.

O'Hara (2011) also combined information from lyrics with another source, in this case chord symbols. He investigated the association between chords and lyrics using a collection of songs which had both lyrics and chord symbol annotations. By associating words with moods via the CAL500 collection, typical moods for chords could be discerned. By generalising the process to sequences of four chords, the mood of the sequence could also be inferred. This work was interesting from our perspective as it was concerned with harmony, and in particular harmonic progressions, together with their relation to text. In O'Hara's case, the text was the song lyric; in our case, the text is that of the question in the C@merata task, which of course is closely back-related to text passages in real musicological documents (Sutcliffe et al. 2015a).

Very interesting and relevant work concerned with text documents and MIR has been carried out by Kuribayashi et al. (2013, 2015). The aim of Kuribayashi et al. (2013) was to develop a method to retrieve content descriptions of classical music

[8] http://guidolib.sourceforge.net/GUIDO/.

such as 'The final section, Nachtwandlerlied, makes subtle use of tonal and thematic cues', without retrieving other spurious texts which are about the music in question but which do not describe it in technical terms. They carried out an analysis of 1540 web pages, retrieved using classical music titles, and found that they could be classified by eight non-exclusive features: Structure (e.g. specific descriptions of part of a composition as above), Background (e.g. descriptions of the composer and aims of the composition), Commentary (evaluation of the composition or a performance of it), Score (scores for sale or available for download), CD/MP3 (recordings for sale or download), NonEng (pages not in English), Dictionary (articles containing only simple descriptions) and Irrelevant (pages that did not fit the labels above). They developed four methods which were intended to retrieve documents which possessed the Structure feature: Technical Term Frequency based Ranking (TTFR), Latent Dirichlet allocation based Ranking (LR), Labelled Latent Dirichlet allocation (LLRC), and Labelled Latent Dirichlet allocation with additional Wikipedia training data (LLRCW).

TTFR scores a document on the proportion of musical technical terms in it (using the Wikipedia pages "Glossary of musical terminology" and "Symphonies"). LR assumes that pages containing good descriptions use similar vocabulary. LDA was applied to Wikipedia pages about symphonies and a latent topic for content descriptions was manually determined from this. Other pages could then be classified based on whether they also featured this latent topic. LLRC is a trained version of LR. The 1540 pages had been assigned the features Structure, Background etc. as above and these were used for training. Pages were then ranked based on how close their word distribution was to that corresponding to the Structure label. Finally, LLRCW used the same method as LLRC, but augmented the training data with further Wikipedia pages.

The approaches were evaluated based on 50 web pages retrieved by a standard search engine for each of ten classical music compositions. Each page was manually scored 0–3 based on its content description and this was used to compare the four ranking methods using Normalised Discounted Cumulative Gain. LLRC and LLRCW performed better than the other two methods and much better than the search engine baseline.

In the conclusion, the authors proposed to link content descriptions to parts of a recording and this appears to be the earliest reference to the idea which we developed further at the end of the same year.

Following their 2013 work, Kuribayashi et al. (2015) presented a method for aligning musical content descriptions obtained from different sources. They observed that partial content descriptions in a paragraph tend to be ordered chronologically for the composition in question, i.e. an earlier musical passage will tend to be discussed before a later one. This suggested the use of sequence alignment techniques for matching descriptions.

Passage expressions were obtained by using their earlier ranking method to select the top 100 paragraphs for each of 23 compositions, resulting in 2300 paragraphs. An initial list of fourteen nouns and 29 verbs was prepared; based on a syntactic analysis of the paragraphs which identified subject, verb and object, sentences containing one of the nouns and one of the verbs were extracted. Where the noun

was subject, the corresponding object was also extracted and vice versa. Where these additional words did not meet the following conditions, they were eliminated: Words must be present in the LLDA training data (i.e. they are in the topic model) and they must be similar to existing seed nouns, defined in terms of word2vec distance (Mikolov et al. 2013).

For Kuribayashi et al. to align paragraphs using Dynamic Time Warping (e.g. Lavecchia et al. 2007) a distance measure for two sentences was required. First, they used the distance between the distribution of word meaning vectors for each sentence. Second, they used the cosine similarity between the meaning vectors for each pair of passage expressions, one in the first sentence and the other in the second. Results were evaluated in terms of P, R and F, using 135 sentences extracted manually from the top 100 paragraphs, ranked using their earlier methods, for ten compositions. There were 41 matching pairs and the best F measure in their trials was 0.632.

Finally, concerning work relating to the C@merata evaluations, there has been work on NLP Information Extraction techniques applied to documents about music. Tata and Di Eugenio (2010) describe a method to extract information about individual songs from reviews which cover the entire album. They identified song titles and used these to find text segments which described that song. WordNet[9] was then used to identify keywords relating to musical features. Following application of the Stanford Typed Dependency Parser, sentences were subdivided into 'f-sentences' so that each portion related only to one feature. F-sentences were then grouped by sub-feature and polarity, in the latter case using SentiWordNet.[10] Finally, a review for each song was assembled from the relevant f-sentences.

Oramas et al. (2014) extracted artist biographies from Grove Music Online. They applied the tokeniser from the Stanford Parser,[11] linked named entities using DBpedia Spotlight,[12] determined the gender of a named entity using GenRe and extracted relations using the MATE parser.[13] Hence a knowledge graph of entities and relations between them was created.

Oramas et al. (2015) present various methods for comparing artists via their biographies. Information was extracted from Last.fm[14] using entity linking with Babelfy[15] and dependency parsing using MATE, followed by the construction of a semantic graph based on the data. Finally, various artist comparison methods based on the graph are evaluated.

Oramas et al. (2016a) present an NLP pipeline for constructing a music knowledge base. They used the Songfacts[16] website as test data for the experiments. The knowledge base holds information about which relations hold between which

[9] http://wordnet.princeton.edu/.

[10] http://sentiwordnet.isti.cnr.it/.

[11] http://nlp.stanford.edu/software/lex-parser.shtml.

[12] http://www.dbpedia-spotlight.org/.

[13] http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/matetools.en.html.

[14] http://www.last.fm/.

[15] http://babelfy.org/.

[16] http://www.songfacts.com/.

named entities. e.g. “Born in the USA” “was recorded by” “Bruce Springsteen”. Relations are on a hierarchy so that more specific ones “was recorded by frontman” can be grouped into less specific ones “was recorded by”. They used the Stanford NLP tokeniser and the MATE dependency Parser. DBpedia Spotlight was used for entity linking along with domain-specific heuristics for co-reference resolution (e.g. “this album”, “this song”). Named Entity types were restricted to Musical Artists, Other Artists, Songs, Albums, Genres, Films and Record Labels. A relation pattern consists of all words in the shortest path between two recognised Named Entities. They employed heuristics for filtering out the useful relations and used dependency-based loose clustering to group them together. Relations were scored using three statistical measures: degree of specificity, intrinsic features and smoothing factor. They carried out experiments evaluating the different components of their IE pipeline: quality of entity linking, quality of relations, coverage of KB and Interpretation of music recommendations. Finally, they carried out an experiment comparing music recommendations made alone, made with a text explanation, and made with an explanation derived from relation patterns used in the recommendation. Their last experiment suggested that explanations in recommender systems could improve the user experience.

Oramas et al. (2016b) combined the output of three standard entity-linking tools, DBpedia Spotlight, TagMe[17] and Babelfy, working with data from Last FM, to produce more accurate data concerning artists, bands and albums.

Oramas and Sordo (2016) is a revised version of Oramas et al. (2014). They created a knowledge graph from information extracted from biographical entries in Grove Music Online. They used standard tools to do this, such as DBpedia Spotlight and TagMe. A comparison was also made with another knowledge based called FlaBase,[18] created by the same authors. The latter is constructed from multiple sources, which can improve accuracy.

Oramas et al. (2016c) describes the creation of a dataset of 65,000 albums constructed from multiple sources, namely Amazon reviews, MusicBrainz[19] and AcousticBrainz.[20] Once again, they performed named entity linking and disambiguation on the texts using TagMe as well as opinion and sentiment analysis. The dataset also included other information such as acoustic features. They carried out various experiments on the data, including review genre classification and a study of how the sentiment associated with a review changes over time.

Oramas et al. (2016d, 2017a) are two tutorials in which the authors have presented their work and explained concepts within NLP and MIR.

Oramas et al. (2017b) present a dataset (MuMu) of 31,000 albums classified into 250 genres. Data includes the image of the cover, reviews in text form, and audio recordings. MuMu combines the Amazon Reviews dataset[21] and the Million Song

[17] http://tagme.d4science.org/tagme/.

[18] http://www.upf.edu/web/mtg/flabase.

[19] http://musicbrainz.org/.

[20] http://acousticbrainz.org/.

[21] http://jmcauley.ucsd.edu/data/amazon/.

Dataset[22] by using MusicBrainz IDs. They then used a Convolutional Neural Network with four convolutional layers to assign one or more genre labels to each album. This was then trained and evaluated with text-based, image-based and audio-based input, as well as different combinations of these. An approach using all three yielded the best results. Oramas (2017) is a thesis which combines much of the above work and includes some useful additional literature review.

Hu et al. (2017) presents an analysis of 53,648 email messages from the Music Library Association Mailing List for the period 2000–2016. Topic Modelling using Latent Dirichlet Analysis (LDA) was used to assign the messages to 27 subjects. The results were then used to track changes in topic popularity over time and to determine topics where a large number of replies were likely.

Tsaptsinos (2017) analyses lyrics using a Hierarchical Attention Network (HAN) in order to classify songs by genre. The work builds on that of Yang et al. (2016) who applied a HAN to the task of text classification. The approach exploited the fact that documents have a structure; words form sentences, sentences form paragraphs and so on. Allowing the network to pay attention to these different levels resulted in a superior classification performance.

Tsaptsinos follows Yang et al. and applies bidirectional Recurrent Neural Networks which use Gated Recurrent Units with attention applied to the output. Such BiGRU networks can capture longer dependencies in the input by virtue of the attention mechanism. First, a BiGRU is applied to an input line, with attention paid to the words; the object is to find which words contribute most to genre classification. A sum of the hidden states weighted by attention is then used in the next BiGRU layer which is applied to a series of lyric lines, with attention paid to lines; here the object is to find which lines contribute to genre classification. Once again a sum of the hidden states weighted by attention is created. Finally, classification into genre is carried out via a softmax activation function applied to the sum. During training, the networks learn the weights including those relating to attention.

A set of 495,188 lyrics derived from LyricFind[23] was used for training, classified in 117 genres, along with a subset of 449,458 lyrics classified into twenty genres. one-hundred-dimensional GloVe[24] word embeddings (distributed representations) were created via the 30,000 most frequent words from LyricFind. In tests following training, the HAN model was the best performing on the 117-genre classification task; for the twenty-genre task, a simpler Long Short Term Memory (LSTM) model was slightly better. Overall, the Tsaptsinos work demonstrates the application of RNNs with attention to an NLP task in the music domain.

Concerning grammatical analysis on music texts, a parser for analysing C@merata question data has been developed by combining a very detailed musical technical terminology with the SpaCy[25] parser (Sutcliffe and Liem 2017; Sutcliffe et al. 2017).

[22] http://labrosa.ee.columbia.edu/millionsong/.

[23] http://lyricfind.com.

[24] http://nlp.stanford.edu/projects/glove/.

[25] http://spacy.io/.

This concludes our discussion of related work. For surveys of MIR in general, the reader is referred to two excellent reviews by Orio (2006) and Schedl et al. (2014), as well as to the MIREX evaluations (Downie 2008).

3 Design of task

3.1 Score format

A fundamental decision for the project was the choice of format for the music scores. Originally we had considered Musical Instrument Digital Interface (MIDI)[26] because it is very well established and relatively simple; moreover there is a huge choice of publicly-available scores in this format. On the other hand, there are considerable disadvantages. MIDI really only captures the sound resulting from the music. There is only an indirect link between a MIDI file and the score from which it was produced (if indeed there was one at all). Some MIDI files can be used to create reasonably accurate scores while others cannot. In consequence of these considerations, the idea of using MIDI was abandoned.

Other contenders were Kern (Huron 1997, 2002), MusicXML[27] and Music Encoding Initiative (MEI).[28] The Kern format was developed at Stanford by David Huron. It is an ASCII format which has been carefully designed and very well tried and tested. Moreover, there is a considerable library of scores available from Stanford in this format.

MusicXML was developed with the intention of capturing most aspects of a music score in a relatively simple and compact XML-based language. It has been widely accepted among developers of commercial score writing tools such as Sibelius and Finale as an interchange format. In consequence, such programs can both import and export in the MusicXML format. This means that anyone who transcribes a public-domain score using a mainstream package can then export the result into MusicXML and make it publicly available for others to use. This is a great advantage for projects such as ours which require a large selection of short scores satisfying a number of different constraints (see section on Development of Data Sets below). Moreover, there is excellent software for MusicXML in the form of the Music21 system from MIT (Cuthbert and Ariza 2010). In addition, anyone with XML processing tools can parse a MusicXML file and extract the information from first principles if they prefer.

Finally, MEI is another XML-based format, influenced by the Text Encoding Initiative (TEI)[29] and aiming to address certain perceived shortcomings of MusicXML from the perspective of the scholarly representation of all aspects of all kinds of scores, including manuscripts. However, there are not many publicly-available scores in this format as yet.

[26] http://www.midi.org/techspecs/.

[27] http://www.musicxml.com/.

[28] http://music-encoding.org/home.

[29] http://www.tei-c.org/index.xml.

Based on the above considerations, it was decided to use MusicXML because it is very widely used, it is supported by many tools resulting in a large number of scores being available, and it can be processed very conveniently within a larger software system (incorporating natural language processing, information retrieval etc.) by Music21. Moreover, scores in Kern or MEI can be converted into MusicXML, albeit with some possible loss of information. The possible shortcomings of MusicXML from a scholarly perspective are not of primary interest to our project since the kind of analysis were are performing on scores and the kinds of information we are asking participants to retrieve are both necessarily fairly simple.

3.2 Evaluation considerations

Evaluations in Information Retrieval normally pose a series of queries, each consisting of a short textual string ('president united states'). The answer is an ordered list of documents which contain the keywords included in the query. Evaluation of the first *n* results returned can be by Precision and Recall (Cleverdon 1962) together with F-Measure which is derived from van Rijsbergen's E-Measure (van Rijsbergen 1979) and is the harmonic mean of Precision and Recall.

Evaluations in Question Answering such as ResPubliQA (Peñas et al. 2009) normally pose a series of textual questions each represented as a short string ('Who is President of the United States?') as we have discussed above. They seek an exact answer mostly comprising an instance of a Named Entity (Mollá and Vicedo 2007) (e.g. 'Barack H. Obama') usually supported by a text snippet drawn from one of the documents in the test collection ('Barack H. Obama is the 44th President of the United States.').[30] Evaluation can be by simple Accuracy (percent of answers correct) (Peñas et al. 2009) or C@1 (Peñas and Rodrigo 2011) which favours a system which declines to answer a question over one which answers it incorrectly. For ResPubliQA, we evaluated all answers manually. In QA4MRE, questions were more complex with answers being multiple choice. This allowed evaluation to be carried out automatically which not only saved time for the organisers but also allowed participants to submit more runs for evaluation; in ResPubliQA we needed to limit the runs to three for practical reasons.

Turning to C@merata, we wished to pose a query in terms of a short natural language phrase ('crotchet F#') and receive in response a series of answers relative to a stated music score. We also needed to evaluate the results by some means analogous to methods in IR and QA. We describe how this was done in the next section.

3.3 Music *passages* in C@merata

Fundamental to our evaluation was the concept of an answer within a score. We considered various types of question each with different answer types, but decided for the first year at least to make questions and answers as simple as possible. A question would simply be a short noun phrase referring to some aspect of a score

[30] www.whitehouse.gov/administration/president-obama.

using the terminology and phraseology of Western classical music. The answer would be a subsection of a score, starting and ending at a particular place.

Given that each stave within a score is almost always divided into bars (measures), it was decided to demarcate an answer primarily in terms of these. An answer would start in a particular bar and end in another (possibly the same) bar.

We also wished to denote the exact starting point of an answer. Initially, we planned to do this in terms of the shortest possible interval of time, i.e. the hemidemisemiquaver (sixty-fourth note), being one sixteenth of a crotchet (quarter note) in length. However, this does not allow for triplets (where, say, a crotchet is divided into three) or any other sort of n-tuplet. This led to the idea of dividing each crotchet into a specified number of beats which would allow the expected answer to be specified exactly.

On further investigation it was discovered that this problem had already been spotted and solved within the MusicXML notation by the concept of *divisions*. The divisions value is the number of beats into which the crotchet is divided. A suitable value depends on what we wish to demarcate as an answer. Working with whole crotchets and nothing smaller, Divisions=1 can be used. Working with quavers (eighth notes) as the smallest time value, Divisions=2 will suffice, while counting in quaver triplets, we can use Divisions=3. Thus to work in semiquavers or quavers or triplet quavers, Divisions=12 is needed, as twelve is the smallest integer divisible by two, three and four. In such a case, one crotchet is twelve beats, one quaver is six beats, one quaver triplet is four beats and one semiquaver (sixteenth note) is three beats.

Thus the divisions concept was adopted in C@merata. MusicXML scores state the divisions value used; normally (but not necessarily) this is the same for each stave and holds true throughout a movement—only scores with this property were used. For simplicity, we specified for each query the divisions value to be used for the answers. This effectively forced participants to specify their answers in the divisions value which we provided—and which we had already checked was sufficient to demarcate all the answers—greatly simplifying the evaluation process.

Based on these ideas we developed the concept of a passage which would contain the following information:

- a start time signature,
- an end time signature,
- a start divisions value,
- an end divisions value,
- a start bar and beat,
- an end bar and beat.

The time signature is denoted in the normal way; thus 4/4 indicates four crotchets in the bar. The start and end time signatures will normally be the same; however, there could be a change of time signature during the passage itself and we are allowing for this.

The start divisions value is the number of beats into which a crotchet in the bar containing the *start* of the passage is divided. Similarly the end divisions value is the number of beats into which a crotchet in the bar containing the *end* of the passage is

divided. As stated above, these were always the same in the first year of the campaign and, because this was found not to pose any problems or restrictions in practical cases, they remained the same for the two subsequent years.

The start bar and beat is where the passage is defined to commence. More precisely, the passage begins in the denoted bar immediately before the start beat, measured from the beginning of the bar in the unit of time denoted by the stated divisions value. Similarly, the passage is defined to end immediately after the end beat. We adopted this before-the-start and after-the-end approach after careful thought and discussion. The advantage of it is that it is intuitive: As we will see below, the first two crotchets in bar 7 can be denoted 7:1–7:2 which can be understood at a glance.

We developed three ways of stating a passage: *ASCII Long Form*, *ASCII Short Form* and *XML form*. These all provide the same information except that the ASCII Short Form assumes both the time signature and the divisions value are the same at the start and end of the passage as we have already mentioned. The ASCII forms are convenient for discussions in papers etc. and for preparing data by hand, something which we did extensively as we lacked the very sophisticated tools which had been developed by Giovanni Moretti and others at CELCT in Trento for our earlier QA tasks at CLEF. The XML form is useful as the input to—and output from—programs. Here is an example in long form:

```
[4/4,4/4,1,1,1:1-2:4]
```

The passage starts and ends in 4/4. The divisions value for the start bar is 1 and so is the divisions value for the end bar. Thus the start bar and the end bar are both divided into four crotchets. The passage starts in bar 1 before the first crotchet (i.e. 1:1) and ends in bar two after the fourth crotchet (i.e. 2:4). In other words the passage consists of the two complete bars numbered one and two.

In the above example, the time signature and divisions value are the same in the start bar as in the end bar, so there is an equivalent short form:

```
[4/4,1,1:1-2:4]
```

The notation is organised so that the passage includes both the beats given. So `[6/8,2,1:4-1:5]` is a passage in 6/8 which comprises the fourth and fifth quavers in the bar. It follows from this that the passage starts before the first beat denoted (1:4) and it finishes after the second beat denoted (1:5). An alternative could have been to define both the start and the end as immediately preceding the beat given. In such a case the very same passage would have been denoted `[6/8,2,1:4-1:6]`, i.e. starting immediately before the fourth quaver beat and ending before the sixth beat. After much debate and consideration, we decided that the first form was more intuitive and easier to use when writing down passages by hand, and when judging the correctness of passages by hand. For automatic processing, either notation would be equally suitable as the same information is being expressed in each case.

As regards the bar numbers, we simply take these from the original MusicXML score, whether or not they are intuitively correct. For example, we would expect the first full bar of a work to be numbered one, with an anacrusis (i.e. incomplete) bar (if

any) being numbered zero. In most cases that is so. However, if it is not, our Gold Standard answers follow the bar numbering found in the score, whatever that may be. One participant over the years has transformed the MusicXML scores into Kern and then searched for results using Kern tools. An unfortunate anomaly came to light whereby, on rare occasions, Kern assigned different bar numbers from MusicXML. This illustrates a typical problem when converting between music formats—it usually works fairly well but is not always completely accurate.

The XML format for a passage essentially followed the ASCII long form and looks like this for `[4/4,4/4,1,1,1:1-2:4]`:

```
<passage start_beats="4" start_beat_type="4"
         end_beats="4" end_beat_type="4"
         start_divisions="1" end_divisions="1"
         start_bar="1" start_offset="1"
         end_bar="2" end_offset="4" />
```

In order to denote a point in the score (e.g. a key change), just one beat can be given, preceded by 'p'. The point is defined to be immediately to the right of the beat given. Thus `[3/4,2,p4:3]` denotes the mid-point of bar number 4 which is in 3/4. The divisions value of two means we are counting in quavers. The point is three quavers from the start of the bar and three quavers from the end. Because the point in the score is after the number of beats specified, we consider this to be equivalent to the end of a passage not the start of a passage. Thus in the XML form for a point we set start_beats, start_beat_type, start_divisions, start_bar and start_offset to the null string '""', while end_bar and end_offset denote the beat which immediately precedes the point being denoted:

```
<passage start_beats="" start_beat_type=""
         end_beats="3" end_beat_type="4"
         start_divisions="" end_divisions="2"
         start_bar="" start_offset=""
         end_bar="4" end_offset="3" />
```

We also specified that the end of a bar should be stated as the start of the following bar with zero offset, i.e. `[3/4,2,p4:6]` should always be written `[3/4,2,p5:0]`.

To conclude our discussion of passages, we would like to comment on some limitations. Firstly, our passage is like two vertical lines drawn through all staves in a score. We do not specify which stave the feature occurred in, by contrast with the proposed system of Viglianti (2015). On the other hand, consider harmonic intervals which in fact were allowed to be across staves in all 3 years of the task (see below). Specifying the stave in a passage would get us into further trouble here, since two staves would genuinely be involved in the same passage.

Secondly, it follows from the first point that two answer passages could overlap. For example, one answer to a particular question could be `[4/4,1,1:1-2:4]` in one stave and another could be `[4/4,1,2:1-2:2]` in another stave. This has actually happened in the task. It is not wrong, but could be regarded as anomalous. In neither case can we tell from the passage specification itself which stave we are talking about.

Thirdly, the passage concept embodies the assumption that the answer we are looking for can be exactly demarcated. What if the demarcation is ambiguous, or if different annotators—each equally qualified—specify a different passage as an answer to a question? In NLP we are quite familiar with this problem (Artstein and Poesio 2008). We can determine inter-annotator agreement concerning a particular piece if information and then not require a system to be more accurate than the ambiguity implied by the differences between expert annotators.

For the last 3 years, we have glossed over the problem of demarcation ambiguity and kept the demarcation of passages (and indeed the overall task) the same because this allows the three Gold Standard datasets to be used interchangeably for developing systems. However, we did discover an important case which seemed to take us to the limit of musical knowledge: cadences. Where does a cadence begin and end? In simple cases it is clear (we chose these for our task) but due to the use of ornaments, broken chords and so forth, there might well be ambiguity. We noted for the future that the instant where the V chord changes to I in the case of a Perfect Cadence is clear if we regard it as denoted by the bass. So maybe a cadence should be a point not a passage. This is exactly where musicology and NLP meet; a musicologist can understand the deep concept of 'cadence'; they know the significance and contribution of stylistic additions such as grace notes but at the same time this does not distract them from a comprehension of the whole. Exactly where the cadence begins and ends is not important to them because they know that different *aspects* of the cadence begin and end at different places in the score.

3.4 Evaluation

As mentioned earlier, Precision, Recall and F-Measure are commonly used in IR and NLP (Cleverdon 1962; van Rijsbergen 1979) while Accuracy and C@1 can be used in Question Answering (Peñas and Rodrigo 2011). In our task, the answer to each question is one or more passages in a score. Our strategy was to determine all the correct answer passages by hand to produce a Gold Standard and then to compare the results returned by a system to that.

Usually one has both strict and lenient measures in an evaluation. For example, at the fourth TREC QA track onwards (starting in 2002) there were four possible judgements of an answer: Right, ineXact, Unsupported and Wrong (Voorhees 2002). In the TREC context, a correct answer could be 'Bill Clinton' while an ineXact one could be 'Clinton' (missing the first name) or perhaps 'Bill Clinto' (end of surname cut off). Unsupported answers were Right but not shown to be so from the document in the collection provided by a participant system to support its answer. For example, if the question is 'Who is President of the US' and (relative to

the historical document collection being used) the answer is 'Bill Clinton', this is Unsupported if the document cited by the system only states 'Bill Clinton was born on 19 August 1946'. This is because the document mentions Clinton but does not state that he was President.

For our task we decided that a passage returned which began at the right bar and beat within the bar and also ended at the right bar and beat within the bar was correct. On the other hand, an answer which started and ended at the right bar (but not necessarily the right beat in the bar) was still very useful and could be considered the equivalent of TREC's inExact. If an expert is looking for a particular type of cadence, for example, and is told the bar numbers, they can see it at a glance. However, searching through hundreds of bars looking for the cadence is time consuming. The concept of Unsupported is not applicable to our task. The measures were thus defined as follows:

We will define *Beat Precision* (BP) as the number of beat-correct passages returned by a system, in answer to a question, divided by the number of passages (correct or incorrect) returned. Similarly, *Beat Recall* (BR) is the number of beat-correct passages returned by a system divided by the total number of answer passages known to exist. As is usual, *Beat F-Score* (BF) is the harmonic mean of BP and BR.

Measure Precision (MP) is the number of bar-correct passages (i.e. measure-correct passages in American terminology) returned by a system divided by the number of passages (correct or incorrect) returned. *Measure Recall* (MR) is the number of bar-correct passages returned by a system divided by the total number of answer passages known to exist. *Measure F-Score* (MF) is the harmonic mean of MP and MR.

The same evaluation measures were used for all 3 years of the task.

4 Development of data sets

4.1 Question types and distribution

For the 2014 evaluation we resolved to devise 200 questions in a carefully crafted distribution. The first step was to work out a set of twelve question types which are shown in Table 2. These vary in their complexity. Queries of type simple_pitch specify the pitch (and possibly octave) of the required note, e.g. 'E', 'B5', 'A natural', 'F#4'. Any note, irrespective of its length, which is of the specified pitch (and octave) is deemed to match. If no octave is specified (e.g. 'E') then any E natural irrespective of octave will match. Ties are ignored in questions of this type and the passage starts at the beginning of the note and finishes at its end.

Queries of type simple_length specify only how long the target note lasts ('dotted quarter note', 'semiquaver rest', 'whole note', 'minim'). Type pitch_and_length combines the first two types ('half note C', 'D# crotchet', 'quarter note B5', 'dotted crotchet A sharp').

perf_spec queries combine a note with some performance information shown in the music notation ('fermata A natural', 'staccato B flat', 'F trill', 'down bow E').

Table 2 C@merata 2014 query types

Type	No.	Example
simple_pitch	30	G5
simple_length	30	dotted quarter note
pitch_and_length	30	D# crotchet
perf_spec	10	D sharp trill
stave_spec	20	D4 in the right hand
word_spec	5	word "Se" on an A flat
followed_by	30	crotchet followed by semibreve
melodic_interval	19	melodic octave
harmonic_interval	11	harmonic major sixth
cadence_spec	5	perfect cadence
triad_spec	5	tonic triad
texture_spec	5	polyphony
All	200	

stave_spec queries restrict the answer to a particular stave in the score which may be specified in various ways (‘half note D in the viola’, ‘D4 in the right hand’, ‘treble clef A sharp’, ‘Bass C#’). word_spec links a note to the word which is sung on it in one of the parts (‘minim on the word "Der"’, ‘eighth note on the word "che"’, ‘word "Se" on an A flat’, ‘G on the word "praise"’). Queries of type followed_by specify two adjacent notes (‘D followed by G’, ‘quarter note G followed by eighth note G’, ‘dotted quaver E followed by semiquaver F sharp’, ‘sixteenth note rest followed by B natural’).

Two query types are concerned with intervals. melodic_interval specifies two adjacent notes on the same stave which are a specified distance apart (‘melodic minor sixth’, ‘rising major sixth’, ‘octave leap’, ‘melodic descending fifth’). Conversely, a harmonic_interval specifies two simultaneous notes at a specified distance (‘harmonic second’, ‘seventh’, ‘minor ninth’, ‘harmonic fifth’). Unlike melodic intervals, harmonic intervals may occur across staves. Intervals mentioned in a query are considered harmonic by default, thus ‘fifth’ is assumed to be a harmonic fifth.

The last three question types were more experimental. cadence_spec requires a cadence to be identified which in 2014 was always perfect (‘perfect cadence’). triad_spec specified triads in various widely-used forms of notation (‘tonic triad’, ‘triad in first inversion’, ‘Ia triad’, ‘Ib triad’). Finally, texture_spec stated the required texture to be found (‘melody with accompaniment’, ‘polyphony’, ‘monophony’, ‘homophony’).

The frequency of queries falling into the various types was fixed based on their complexity, with the simplest query types (simple_pitch, simple_length, pitch_-and_length, followed_by) being the most numerous in the test set with 30 each (Table 2). After this came stave_spec and melodic_interval with twenty each followed by perf_spec and harmonic_interval with ten each. (One melodic_interval

was changed for a harmonic_interval at a late stage, so there were nineteen of the former and twenty of the latter. Finally, there were five each of word_spec, cadence_spec, triad_spec and texture_spec.

Based on our experience in 2014, there were two conclusions regarding the questions. Firstly, a lot of the queries were rather basic, with answers that were not that difficult to find and not particularly interesting. Secondly, the system of question types restricted what questions could be asked and in particular it limited the ways in which certain types of restriction could be placed on the desired passage. As a result, we adopted a different strategy for 2015. The questions were now based on six fundamental types: 1_melod, n_melod, 1_harm, texture, follow, and synch (Table 3). Moreover, each of these could be modified in five different ways, perf, instr, clef, time and key. Finally, we could restrict any query to a range of bars (measures).

A 1_melod query can be a note name, a note length, or the two combined (e.g. 'F#6', 'dotted half note', 'dotted minim F#4'). It thus combines the first three query types from the first year. A 1_melod can be modified by performance style (perf) ('trill on a quarter note C'), instrument (instr) ('eighth note E5 in the Horn 2'), clef ('sixteenth note C# in the left hand'), time signature (time) ('whole note C5 in 4/4') and key ('dotted quarter note A4 in F major').

An n_melod query can be a specified number of notes ('four quaver C4', 'five note melody'), a melody or sequence specified in note names or Tonic Sol-Fa ('F# E G F# A', 'Do Mi Do Sol Do Mi Sol Do'), lengths ('crotchet, crotchet rest, crotchet rest') or both ('32nd note E, 32nd note F#'), a scale or arpeggio ('descending 8-note scale of C major in half notes', 'descending arpeggio in eighth note triplets') or a melodic interval ('melodic octave leap'). Once again, n_melod queries can be modified, for example, by instr ('eight Bb5 in the Oboe'), clef ('six minims in the bass clef'), time ('G4 B4 E5 in 3/4') or key ('G major arpeggio').

1_harm queries specify a harmonic interval ('harmonic minor third') or chord ('chord Eb Ab Bb D F'). They can be modified, e.g. by instr ('harmonic minor third in the Violins 1') or clef ('dotted minim chord in the left hand'). texture questions were similar to 2014 ('monophonic passage', 'homophony').

follow questions link two of the previous types, with or without modification, specifying that one must come first and the other second. Here are some examples: 'dotted quarter note, eighth note, eighth note, eighth note followed by D5' (n_melod_follow_1_melod), 'quarter note B3 followed by whole note chord A3 C#4 E4 A4' (1_melod_follow_1_harm), 'a half note in the Horn 2 followed by a quarter note in the Horn 1' (1_melod_instr_follow_1_melod_instr), 'harmonic octave Eb in the piano left hand followed by harmonic octave F in the piano left hand' (1_harm_clef_follow_1_harm_clef).

synch questions specify two elements which occur at the same time: 'four quavers in the violin against a minim in the bass clef' (n_melod_instr_synch_1_melod_clef), 'six eighth notes against a dotted half note' (n_melod_synch_1_melod), 'quaver chord C4 E4 against a crotchet C5' (1_harm_synch_1_melod), 'eight eighth notes in the Violin I against a B pedal in the Cello' (n_melod_clef_synch_1_melod_clef).

Table 3 C@merata 2015 query types

Type	No.	Example
1_melod	40	D4 minim
		eighth note in measure 9
1_melod qualified by perf, instr, clef, time, key	40	trill on a quaver A
		G# in the Cello part in measures 29–39
		sixteenth note C# in the left hand
		half note E3 in 2/2
		sixteenth note G in G minor in measures 1–5
n_melod	20	F# E G F# A
		Do Mi Do Sol Do Mi Sol Do in bars 1–20
		twenty semiquavers
		five note melody in bars 1–10
n_melod qualified by perf, instr, clef, time, key	20	two staccato quarter notes in the Violin 1
		crotchet, crotchet rest, crotchet rest, crotchet, crotchet rest, crotchet, crotchet, crotchet, crotchet, crotchet in the Timpani
		melodic octave leap in the bass clef in measures 70–80
		G4 B4 E5 in 3/4
		rising G minor arpeggio
1_harm possibly qualified by perf, instr, clef, time, key	20	eighth note chord Bb, C, E
		chord of D minor in measures 109–110
		harmonic minor sixth in the Violas
		dotted minim chord in the left hand
texture	6	monophonic passage
		homophony in measures 1–14
		polyphony in measures 10–14
		Alberti bass in measures 0–4

Table 3 continued

Type	No.	Example
follow possibly qualified on either or both sides by perf, instr, clef, time, key	40	quavers F4 E4 in the oboe followed by quavers E2 G#2 in the bass clef
		quarter note minor third followed by eighth note unison
		C followed by mordent Bb
		chord C4 G4 C5 E5 then a quaver
		three eighth notes in the Violin I followed by twelve sixteenth notes in the Violin II in measures 87–92
synch possibly qualified in either or both parts by perf, instr, clef, time, key	14	four eighth notes against a half note
		crotchet D3 on the word "je" against a minim D2
		four staccato quavers in the Violoncello against a minim chord Ab3 C4 F4 in the Harpsichord
All	200	

Finally, any query can be restricted by bar/measure: ‘quaver C in bars 22–27’ (1_melod), ‘slurred quarter note in measures 76–77’ (1_melod_perf), ‘Bb5, G5, F5, E5, Eb5 in the Oboe followed by two A natural crotchets in the Violins 1 in bars 90–100’ (1_melod_instr_follow_1_melod_instr).

As can be seen, the above taxonomy gives us a much broader range of queries than last year. These are more difficult for participants to answer, of course, but they are also more interesting and potentially more useful. The distribution of question types can be seen in the Table 3. Once again we had 200 queries. There are 40 simple 1_melod, 40 1_melod qualified by perf, instr etc., 20 n_melod and 20 qualified n_melod, 20 1_harm (three qualified, the rest not), six of type texture, 40 of type follow (some with qualifications on either side as illustrated above) and finally fourteen of type synch.

Turning to 2016, the distribution of question types is shown in Tables 4 and 5. These tables also show several examples of each type. All 1_melod questions are concerned with one note and can be modified by bar/measure (‘A#1 in bars 44–59’). Thirty-six of the forty can also be modified by perf (‘forte’), instr (‘in the violin’), clef (‘in the bass clef’), time (‘in 3/4’) or key (‘with G major key signature’).

n_melod questions are concerned with a sequence of notes which can be specified exactly (‘D4 D5 A5 D6 in sixteenth notes’) or inexactly (‘two-note dotted rhythm’). They can also be modified by bar/measure, perf etc. in the same way as 1_melod questions.

1_harm questions deal with single chords (‘whole-note unison E2 E3 E4’) which can be less specific (‘chord of C’ or ‘five-note chord in the bass’). Once again they can be modified as above (‘chord of F#3, D4 and A4 in the lower three parts’, ‘harmonic octave in the bass clef’). Note that we allow two notes to be a chord, including octaves etc. In 2016, references to inversions (‘Ia chord’) were considered of 1_harm type.

n_harm questions deal with sequences of chords with the usual modifications (‘three consecutive thirds in bars 1–43’). Cadences are also included here, since they are sequences of specific chords (‘plagal cadence in bars 134–138’). There are also some more complex types (‘A5 pedal in bars 116–138’).

Finally, there are texture questions (‘all three violin parts in unison in measures 1–59’, ‘counterpoint in bars 1–14’). Some more complex forms were added this year (‘imitative texture in bars 1–18’).

Table 5 shows two further forms of question, follow and synch. There are twenty of the former and thirteen of the latter. In a departure from 2015, these are not separate types, but range over the queries of type 1_melod, n_melod, 1_harm and n_harm as shown in Table 4. Thus the examples in Table 5 are all within the distribution of query types shown in Table 4. A follow question allows us to specify some passage followed by another passage. Each such passage can be of type 1_melod, n_melod etc. This allows quite complex sequences to be specified (‘D C# in the right hand, then F A G Bb in semiquavers in the left hand’, ‘5 B4s followed by a C5’).

Questions of type synch can link two passages which must occur at the same time. In the simplest case, each passage is of exactly the same length (‘quarter note E5 against a quarter note C#3’). However, this is not necessarily the case (‘C#3

Table 4 C@merata 2016 query types

Type	No.	Example
1_melod	4	A#1 in bars 44–59
		quarter-note rest in measures 1–5
1_melod qualified by perf, instr, clef, time, key	36	dotted quarter note D6 in the first violin
		solo C5 in the oboe in measures 32 onwards
		flute dotted half note only against strings
		half note on the tonic in the bass clef
		A4 sung to the word 'bow'
n_melod	15	two-note dotted rhythm in measures 1–24
		eight note rising passage in quarter notes
		repeated Bb4 whole note
		D4 D5 A5 D6 in sixteenth notes repeated twice
		two tied dotted minims in bars 72–88
n_melod qualified by perf, instr, clef, time, key	45	dotted minims C B A in the Bass clef in bars 70–90
		melodic interval of a minor 7th in the voice
		rising arpeggio in the left hand in measures 1–10
		five-note melody in the cello in measures 20–28
		whole note rest, quarter note in the Violin 4 in measures 1–103
1_harm	17	7th triad in measures 1–3
		Ia chord in bars 1–10
		chord of C
		whole-note unison E2 E3 E4
		chord III in bars 44–59

Table 4 continued

Type	No.	Example
1_harm possibly qualified by perf, instr, clef, time, key	23	chord of F#3, D4 and A4 in the lower three parts
		harmonic fifth in the oboe
		harmonic octave in the bass clef
		harmonic perfect fourth between the Soprano and Alto in bars 1–9
		cello and viola playing dotted minims an octave apart in bars 40–70
n_harm	25	interrupted cadence
		A5 pedal in bars 116–138
		authentic cadence in measures 14–18
		plagal cadence in bars 134–138
		three consecutive thirds in bars 1–43
n_harm possibly qualified by perf, instr, clef, time, key	15	consecutive sixths between the Altos and Basses in measures 73–80
		flute, oboe and bassoon in unison in measures 1–56
		consecutive descending sixths in the left hand
		alternating fourths and fifths in the Oboe in bars 1–100
		Soprano and Alto moving one step down together in measures 1–12
texture	20	all three violin parts in unison in measures 1–59
		polyphony in measures 5–12
		homophonic texture in measures 125–138
		imitative texture in bars 1–18
		counterpoint in bars 1–14
All	200	

Table 5 C@merata 2016 follow and synch queries within 1_melod, n_melod, 1_harm and n_harm

Type	No.	Example
follow possibly qualified on either or both sides by perf, instr, clef, time, key	20	C D E F D E C in semiquavers repeated after a semiquaver
		eighth-note twelfth followed by whole-note minor tenth between Cello and Viola
		D C# in the right hand, then F A G Bb in semiquavers in the left hand
		B flat in the cbass followed a quarter note later by B natural in the cbass
		5 B4s followed by a C5
synch possibly qualified in either or both parts by perf, instr, clef, time, key	13	quarter note E5 against a quarter note C#3
		C#3 minim and E4 semibreve simultaneously
		D3 in the bass at the same time as C5 in soprano 1
		three-note chord in the harpsichord right hand against a two-note chord in the harpsichord left hand in measures 45–52
		A#3 in the piano and F#5 in the voice simultaneously

minim and E4 semibreve simultaneously'); here, according to our rules, the whole of the minim must lie somewhere within the duration of the semibreve. The length of the passages need not be specified ('D3 in the bass at the same time as C5 in soprano 1', 'three-note chord in the harpsichord right hand against a two-note chord in the harpsichord left hand in measures 45–52').

When we reach the follow and synch questions, they are starting to become interesting from a musicological perspective as such musical phenomena as these cannot readily be specified except in a natural language. The key advantage of language here is that it can vary in specificity from the constrained to the open; to interpret the open queries requires considerable musical knowledge. These queries are starting to relate more closely to actual examples in detailed technical texts (Sutcliffe et al. 2015a).

4.2 Scores

In each of the three campaigns, twenty music scores in MusicXML were chosen for the task. Ten questions were set on each score. As some English-speaking countries use European terminology (crotchet, bar etc.) while others use the equivalent American terms (quarter note, measure etc.), it was decided to set some questions in each form so that all systems developed would be able to handle both. This was accomplished by posing American queries for half of the scores (i.e. ten) and English queries for the other half.

For the 2014 task, scores were chosen from the Renaissance and Baroque periods. The number of staves and scoring can be seen in Table 6, as well as the language (American/English) used for the queries. The distribution by staves is shown in Table 7. There were six scores containing two staves and six on three staves, four on one stave and two each on four staves and five staves.

Instrumentation is shown in Table 6. Scores on one stave were not necessarily in one part; the Prelude from the first Cello Suite of Johann Sebastian Bach is polyphonic in places, as is the lute Galliard by Francis Cutting. Scores on two staves were either for a keyboard instrument (harpsichord) or for the lute in the case of the Prelude from the Sonata No. 34 of Silvius Leopold Weiss. Scores on three staves were either for a solo instrument and keyboard (e.g. Se Florindo è Fedele by Alessandro Scarlatti) or voices (e.g. SSA in Fie Nay Prithee Z 10 by Henry Purcell). Those on four staves were both chorales (e.g. SATB in the Chorale 24835b3 of Johann Sebastian Bach). Finally, one of the scores on five staves was for instruments (i.e. 2 vn, va, 2 vc in the Preludium from the Te Deum of Marc Antoine Charpentier) while the other was for voices (SSATB in Lasciatemi Morire by Claudio Monteverdi).

The scores were obtained from two sources and all were already in MusicXML (i.e. none were converted by us from other formats such as KERN (Huron 1997, 2002) or MIDI. Most came from musescore.com. This includes a very wide choice but we were looking for particular composers and genres. In addition we required scores to have a license 'to share' rather than just 'for personal use'. Moreover, we required scores to be well presented, transcribed in a scholarly manner and provided in valid MusicXML Version 2 or lower. These constraints

reduced a vast database down to a set of candidate scores which was only just sufficient to choose music for our task. Two Bach Chorales were used and both came from www.jsbchorales.net which includes all the chorales.

In the 2015 task, the range of repertoire was extended to include the Classical and early Romantic periods in addition to the Renaissance and Baroque. Table 8 shows the works chosen, number of staves, instrumentation and query language (American/English). Table 9 shows the distribution of staves at a glance. One of the simplest scores was probably mozart_12_horn_duos_k487_no1_allegro which was on two staves and lasted 34 bars. By contrast beethoven_symphony_no3_mvt1_exposition was on nineteen staves and lasted 149 bars.

Concerning the genres of the music, they included works for solo violin by Bach, harpsichord works by Scarlatti and Bach, piano sonatas by Clementi and Mozart, horn duos by Mozart, a fantasia for viols by Purcell, string quartets by Haydn and Beethoven, a cappella polyphonic vocal music by Monteverdi, Sweelinck and Telemann, songs by Mozart and Schubert, a Vivaldi violin concerto, a Bach Brandenburg concerto, and extracts from symphonies by Mozart and Beethoven. Taken together, the scores encompassed a range of music styles and instrumental combinations. They also varied in their rhythmic, melodic and harmonic complexity from straightforward to highly sophisticated.

Almost all scores came from musescore.com, as was the case in 2014. All were already in MusicXML. We chose scores which had a license 'to share' rather than just 'for personal use'. We aimed to select scores which were well presented, transcribed in a scholarly manner and provided in valid MusicXML Version 2 or lower. We tested all scores in the MuseScore[31] software and rejected those which crashed MuseScore (as many did). Once again, it was hard to choose scores which satisfied all our criteria and also lay on our required distributions of staves and so on. However, the wider range or periods and musical genres helped increase the potential pool.

Finally, in 2016, the third year of the task, scores were chosen from the Renaissance, Baroque, Classical and Early Romantic periods. The twenty MusicXML scores were selected from kern.ccarh.org and from musescore.com. The former scores are from Stanford and have been prepared from various public domain and out-of-copyright sources. They are created in the kern format and are also available in a conversion to MusicXML. Some of the Stanford conversions to MusicXML were suitable for our use, but others caused problems in MuseScore. In the latter case, we tried re-converting from kern using music21 and often this produced a better version of the score. In 2016, we relaxed the condition that scores be in MusicXML Version 2 or lower and we allowed Version 3 scores as well. This did not appear to cause a problem with the participants' systems, and such scores are very well supported by music21.

The scores can be seen in Table 10, which shows the work, number of staves and scoring. Composers this year were Bach, Beethoven, Bennet, Chopin, Handel, Morley, Mozart, Palestrina, Scarlatti, Schubert, Vivaldi and Weelkes. There were six works for keyboard (three for harpsichord and three for piano), one Schubert

[31] http://musescore.org/.

Table 6 C@merata 2014 scores

Work	Staves	Scoring	Lang
bach_cello_suite_1_bwv1007_prelude	1	vc	Amer.
bach_chorale_24835b3	4	SATB	Eng.
bach_chorale_507b	4	SATB	Amer.
bach_minuet_in_g_bwv_anh114	2	hpd	Amer.
carissimi_o_felix_anima	3	SAB	Eng.
charpentier_te_deum_preludium	5	2 vn, va, 2 vc	Amer.
corelli_allegro_tr_clementi	2	hpd	Eng.
cutting_galliard_11	1	lute	Eng.
dowland_earl_of_essex_measure	1	A	Amer.
lassus_psalm_50	3	SAB	Eng.
lully_andante	3	2 vn, vc	Amer.
monteverdi_lasciatemi_morire	5	SSATB	Amer.
purcell_fie_nay_prithee_zd10	3	SSA	Eng.
scarlatti_a_se_florindo	3	S, hpd	Amer.
scarlatti_k466	2	hpd	Eng.
tallis_all_praise_to_thee	1	A	Eng.
telemann_taenzchen	2	hpd	Amer.
telemann_twv33_21_tres_vite	2	hpd	Eng.
vivaldi_concerto_rv299_largo	3	vc, hpd	Eng.
weiss_sonata_34_prelude	2	lute	Amer.

Table 7 C@merata 2014 distribution of scores by number of staves

Staves	Frequency
1	4
2	6
3	6
4	2
5	2
All	20

song for voice and piano and two string quartet movements; there were three A Cappella vocal works for SATB and one each for SATTB and SSATB; there were two Vivaldi concertos for strings and continuo and two symphony movements, one by Mozart and the other by Beethoven. Finally there was a movement from HandelÇÖs Messiah for SATB and orchestra.

The distribution of scores in terms of staves can be seen in Table 11. It is similar to 2015 except that there are now five scores with eight or more staves (rather than two): two on eight staves and one each on ten, thirteen and eighteen staves.

Table 8 C@merata 2015 scores

Work	Staves	Scoring	Lang
bach_violin_sonata_no1_bwv1001_presto	1	vn	Amer.
bach_violin_sonata_no2_bwv1003_andante	1	vn	Eng.
mozart_piano_sonata_k545_m1	2	pf	Amer.
mozart_12_horn_duos_k487_no1_allegro	2	2 hn	Amer.
bach_cpe_duet_flute_violin_h598_andante	2	fl, vn	Amer.
clementi_sonatina_op36_no1_v2	2	pf	Eng.
scarlatti_k30	2	hpd	Eng.
bach_marcello_bwv974_adagio	2	hpd	Amer.
mozart_an_chloe_k526	3	S, pf	Amer.
telemann_ceciderunt_in_profundum	3	SSB	Amer.
purcell_fantasia_no10_4_parts_z741	4	4 viols: tr, 2 t, b	Eng.
bach_brand_conc_no2_bwv1047_andante	4	fl, ob, vn, bc	Eng.
haydn_str_quartet_op74_no1_menuetto	4	2 vn, va, vc	Amer.
beethoven_str_quartet_op18_no1_adagio	4	2 vn, va, vc	Amer.
sweelinck_miserere_mei	4	SATB	Eng.
monteverdi_ave_maris_stella	4	SATB	Amer.
schubert_an_die_sonne_d439	6	SATB, pf	Eng.
mozart_symphony_no1_mvt1_exposition	7	2 ob, 2 hn, 2 vn, va, vc, db	Eng.
vivaldi_vn_conc_f_min_rv297_allegro	8	solo vn, 2 vn, va, vc, cb, hpd	Eng.
beethoven_symphony_no3_mvt1_exposition	19	2 fl, 2 ob, 2 cl, 2 bn, 3 hn, 2 tpt, timp, 2 vn, va, vc, db	Eng.

Table 9 C@merata 2015 distribution of scores by number of staves

Staves	Frequency
1	2
2	6
3	2
4	6
6	1
7	1
8	1
19	1
All	20

4.3 Creation of questions

The general approach to the preparation of queries remained the same for the 3 years 2014–16. Each score was sent to one of the organisers who was asked to set questions according to the target distribution for that year (see Tables 2, 3, 4, 5). It was specified for each score whether the questions were to be in American or English (Tables 6, 8, 10). For each question, answers were to be provided in the ASCII short form for specifying passages (see earlier discussion). The organiser in question was asked to find all answers to each question. The question data was returned in an ASCII format which incorporates the score filename, the questions, the answers in ASCII form and also any comments concerning the questions or answers.

On receipt of the files, the questions and answers were checked by a second expert who noted any changes or observations using comments in the ASCII file. The second expert also carried out an independent search for answer passages within the scores. When all changes were checked and validated, the complete set of twenty ASCII files, one for each score, was then transformed automatically into XML format in order to form the Gold Standard for that year's task.

The detailed aspects of question generation did vary, especially in the third year. In 2014 and 2015, we decided the question types and distribution first; we then selected scores and devised queries by going through them, trying to find passages against which queries could be posed. This reverse-logic approach was a simple development of the one which we had used at CLEF for many years (Peñas et al. 2009, 2010, 2011, 2012a, b, 2013). In 2016, we aimed to generate some of the questions using a more realistic approach. Two suggestions had been made in previous years. The first was to base certain questions on First Species Counterpoint as exemplified for example in Kitson (1907) and indeed Fux (1725). In particular:

- *Modes* Dorian, Phrygian, Lydian, Mixolydian, Aeolian, Locrian, Ionian (these would be n_melod queries);
- *Melodic intervals* diminished fifth, augmented fourth (n_melod queries);

Table 10 C@merata 2016 scores

Work	Staves	Scoring	Lang
bach_2_part_invention_no1_bwv772	2	hpd	Eng.
beethoven_piano_sonata_no2_m4	2	pf	Amer.
beethoven_piano_sonata_no5_m1	2	pf	Eng.
chopin_prelude_op28_no15	2	pf	Eng.
scarlatti_sonata_k281	2	hpd	Eng.
scarlatti_sonata_k320	2	hpd	Amer.
schubert_an_die_musik_d547	3	S, pf	Amer.
bach_chorale_bwv347	4	SATB	Amer.
beethoven_str_quartet_op127_m1	4	2 vn, va, vc	Eng.
bennet_weep_o_mine_eyes	4	SATB	Eng.
handel_water_music_suite_air	4	2 vn, va, vc	Amer.
palestrina_alma_redemptoris_mater	4	SATB	Amer.
schubert_str_quartet_no10_op125_d87_m3	4	2 vn, va, vc	Eng.
morley_now_is_the_month_of_maying	5	SATTB	Eng.
weelkes_hark_all_ye_lovely_saints	5	SSATB	Eng.
vivaldi_conc_4_vns_op3_no10_rv580	8	4 vn, 2 va, vc, db	Amer.
vivaldi_conc_vn_op6_no6_rv239_m1	8	3 vn, va, vc, db, hpd	Amer.
mozart_symphony_no40_m4	10	fl, 2 ob, 2 bn, 2 hn, 2 vn, va, vc, db	Eng.
beethoven_symphony_no3_m3	13	2 fl, 2 ob, 2 cl, 2 bs, 2 hn, 2 tpt, timp, 2 vn, va, vc, db,	Amer.
handel_messiah_and_the_glory	18	fl, 2 ob, cl, bs, hn, tbn, tuba, SATB, hpd, 2 vn, va, vc, db	Amer.

Table 11 C@merata 2016 distribution of scores by number of staves

Staves	Frequency
2	6
3	1
4	6
5	2
8	2
10	1
13	1
18	1
All	20

- *Harmonic intervals* perfect concords, imperfect concords and discord (1_harm queries);
- *Movement of parts* similar, contrary, oblique and parallel (n_melod against n_melod);
- *Special relationships* false relation of the tritone (1_harm)
- Exposed fifths and octaves (n_harm).

In the event, we managed queries relating to melodic intervals and movement of parts and we plan to investigate the others in future. The second suggestion was to base questions on music exam papers set in English schools at GCSE level (aged sixteen) and A level (aged eighteen).

In 2016, we did indeed generate some questions based on a study of exam papers. EDEXCEL and AQA GCSE and A-level papers were studied, looking for questions which were suitable for use in the task and which were concerned with pieces of music whose scores were publicly available. A question, once found, then had to be converted into a suitable form for use in the C@merata task.

For example, the AQA A2 Music in Context question paper (Unit 4) from June 2015, question 3c, reads:

> Which one of the following chromatic chords is used in the piano accompaniment to the word 'eine' in bar 15? Underline your answer:
> augmented sixth
> diminished seventh
> Neapolitan sixth
> secondary seventh

It was necessary to convert from a multiple-choice form of question to one based on passages. In this case, the mention of the lyric was also removed to make the question easier, resulting in the following C@merata query:

> diminished seventh in the piano in bars 10–15
> answer: bar 15, minim beat 2

Table 12 C@merata 2014 query and answer distributions

Type	No.	Shortest query	Longest query	Avg. query	Min ans	Max ans	Avg. ans
simple_pitch	30	1	2	1.200	1	20	7.267
simple_length	30	1	3	2.167	1	90	10.833
pitch_and_length	30	2	4	2.867	1	21	5.267
perf_spec	10	2	6	3.700	1	7	2.900
stave_spec	20	2	8	5.000	1	9	3.750
word_spec	5	5	7	5.800	1	6	2.400
followed_by	30	4	8	5.967	1	14	4.267
melodic_interval	19	1	4	2.474	1	24	6.474
harmonic_interval	11	1	3	2.000	1	27	6.364
cadence_spec	5	2	2	2.000	1	3	1.600
triad_spec	5	2	4	2.400	1	6	2.200
texture_spec	5	1	3	1.800	1	4	1.600
All	200	1	8	3.160	1	90	5.825

Other exam questions used were more open-ended and were not multiple choice, sometimes requiring single specific answers or at other times inviting the student to apply relevant terms of their choice in describing a passage. These were converted in an analogous way. Naturally, our referring expressions are only one of innumerable methods by which musicologists refer to music in text. However, our very restricted approach has served us well in establishing an approach for an evaluation task.

The third strand of query creation work was concerned with the derivation of queries from musicological texts. For this campaign, we re-visited some of the texts we studied previously (Sutcliffe et al. 2015a) and styled some of the more complex questions accordingly.

Tables 12, 13, 14 and 15 give some statistics concerning the queries for the 3 years. The length of a query can give a simple measure of its possible complexity. In 2014 (Table 12) queries ranged between one and eight tokens in length, not including punctuation. The average length was 3.160 tokens. By 2015 (Table 13) these figures had increased to 1, 22 and 7.230. Thus queries had more than doubled in length on average. In 2016 (Table 14), queries ranged between 6 and 22 tokens and had an average length of 8.695. So there was a slight increase in query length overall.

In 2014, the shortest type of question was simple_pitch (average length 1.200, Table 12). This was generally a very simple note specification like ‘C#’. Questions of type followed_by were the longest in that year (5.967 tokens on average).; This is to be expected, as such queries must specify two musical features (e.g. ‘crotchet’, ‘semibreve’) and also link them together (‘crotchet followed by semibreve’). In 2015, 1_melod queries (the equivalent of simple_pitch) were also the shortest at 2.900 tokens on average. Queries of type follow and synch were the longest (11.750 tokens and 11.929 tokens on average, Table 13). Once again, these are linking two

Table 13 C@merata 2015 query and answer distributions

Type	No.	Shortest query	Longest query	Avg. query	Min ans	Max ans	Avg. ans
1_melod	40	1	6	2.900	1	8	3.000
1_melod_perf	17	2	8	4.824	1	6	1.824
1_melod_instr	10	5	9	6.600	1	9	3.800
1_melod_clef	6	6	12	9.000	1	7	2.667
1_melod_time	6	5	9	6.167	1	6	2.333
1_melod_key	1	10	10	10.000	10	10	10.000
n_melod	20	2	12	5.650	1	7	2.800
n_melod_perf	0	0	0	0.000	0	0	0.000
n_melod_instr	7	5	16	10.143	1	3	1.714
n_melod_clef	9	6	12	9.000	1	4	2.333
n_melod_time	3	5	7	5.667	1	4	2.333
n_melod_key	1	4	4	4.000	6	6	6.000
1_harm	17	3	12	6.529	1	9	2.235
1_harm_instr	3	6	7	6.333	1	10	4.000
texture	6	2	6	4.667	1	2	1.167
follow	40	5	22	11.750	1	5	1.550
synch	14	7	18	11.929	1	8	2.571
All	200	1	22	7.230	1	10	2.430

Table 14 C@merata 2016 query and answer distributions. In 2016, follow and sych questions were within the counts for 1_melod etc

Type	No.	Shortest query	Longest query	Avg. query	Min ans	Max ans	Avg. ans
1_melod	40	2	18	9.300	1	15	3.400
n_melod	60	3	26	10.250	1	10	2.350
1_harm	40	2	22	8.275	1	20	4.150
n_harm	40	2	13	7.175	1	5	1.775
texture	20	2	12	6.700	1	4	1.750
All	200	2	26	8.695	1	20	2.745
follow	20	5	17	11.400	1	10	2.150
synch	13	6	22	13.154	1	12	3.692

separate passage specifications. In 2016, 1_melod questions had become complex, their average length having risen to 9.300 (Table 14). Texture questions were now the shortest on average (6.700 tokens). Some of the n_melod queries in 2016 were very long, as they specified lists of notes which had to be matched. Hence the longest n_melod was 26 tokens and the average 10.250 tokens (Table 14). Also in 2016, follow and synch queries were distributed over other query types. We can see

Table 15 C@merata 2014 participants

Runtag	Leader	Affiliation	Country
CLAS	Stephen Wan	CSIRO	Australia
DMUN	Tom Collins	De Montfort University	England
OMDN	Donncha Ó Maidín	University of Limerick	Ireland
TCSL	Nikhil Kini	Tata Consultancy Services	India
UNLP	Kartik Asooja	NUI Galway	Ireland

Table 16 C@merata 2015 participants

Runtag	Leader	Affiliation	Country
CLAS	Stephen Wan	CSIRO	Australia
DMUN	Tom Collins	De Montfort University	England
OMDN	Donncha Ó Maidín	University of Limerick	Ireland
TCSL	Nikhil Kini	Tata Consultancy Services	India
UNLP	Kartik Asooja	NUI Galway	Ireland

Table 17 C@merata 2016 participants

Runtag	Leader	Affiliation	Country
DMUN	Andreas Katsiavalos	De Montfort University	England
KIAM	Marina Mytrova	Keldysh Institute of Applied Mathematics	Russia
OMDN	Donncha Ó Maidín	University of Limerick	Ireland
UMFC	Paweł Cyrta	Fryderyk Chopin University of Music	Poland

in the last two rows of Table 14 that these questions were the longest overall (11.400 tokens and 13.154 tokens on average, Table 14). Overall, then, queries were becoming longer from year to year, with the biggest increase from 2014 to 2015.

Turning now to the number of answers to a query, we see the opposite trend, one of decrease rather than increase. In 2014, the average number of answer passages for a query was 5.825 (Table 12) and some queries had a considerable number of answers. For example, one of the simple_length queries had 90 answers. This query was searching for a note length which occurred very commonly in the score in question. On the other hand, at least one query of every type in 2014 had just one correct answer passage. Generally, however, 2014 queries were easy and had many answers. In 2015, the average number of answer passages had fallen to 2.430 (Table 13). This was because queries had become more specific and were matching less in consequence. In 2016, the average number of passages remained similar at 2.745 (Table 14).

5 Campaigns

5.1 Organisation and participation

The organisation of the campaigns was very similar each year. C@merata was one of the tasks at MediaEval (Larson et al. 2014, 2015; Gravier et al. 2016). In 2014, five participants registered for the task, as shown in Table 15. Two were from Ireland and the other three came from Australia, England and India. In 2015, the very same five registered (Table 16). For 2016, there were four participants, one each from England, Ireland, Poland and Russia (Table 17).

Participants had 1 week to complete their runs starting from 16th June 2014, 15th June 2015 and 13th June 2016. They were asked to download the test questions, run them through their systems and upload the results for evaluation to the MediaEval website within 72 h of their download. Each participant was allowed to submit up to three runs.

5.2 Participant systems

Each year, the participants built or augmented a system for carrying out the C@merata task. Originally, we provided a baseline system (Sutcliffe 2014) which participants were free to use if they wished. This system first read the queries in the specified XML input format. Queries were parsed with the Stanford parser (Socher et al. 2013), and were assigned to a question category in the traditional QA fashion. The scores were read in using music21 (Cuthbert and Ariza 2010). For one or two categories, code was provided which could search for a matching portion in the score, represented as a music21 object. The matching part was converted to a C@merata passage specification. Finally, the answers were written out in the required XML format. The Basline System was thus a basic end-to-end solution to the task, performing all the basic operations but minus the detailed algorithms for answering the queries properly. It was written in Python 2.7.

The participants for each year are listed in Tables 15, 16 and 17. Merging the runtags for the 3 years, we have CLAS, DMUN, KIAM, OMDN, TCSL, UMFC and UNLP. DMUN and OMDN have participated in all 3 years; CLAS and UNLP took part twice; KIAM and UMFC joined in the 2016 task.

There are four stages in the 2014 CLAS approach (Wan 2014a, b), which used Music21 as a starting point. First, the input noun phrase comprising a list of words is transformed into a concept representation comprising a list of concepts. Multi-word entities which must be joined (e.g. down bow -> down_bow), compounds which must be separated (Vb -> V b) and quotations (e.g. the word “praise”) are found at this stage. A concept comprises a musical object, an attribute of that object and a value.

Second, concepts are identified in the list which define the type of the answer (e.g. cadence) and the form of music data score to be searched. For example, when searching for a cadence, a chordal form of the score is used. On the other hand, the default form is the concatenation of the sequence of notes in each voice. Third, the

concept list is parsed using a hand-crafted grammar, producing a query representation. So, for example, the concepts underlying '2', 'dotted' and 'crotchets' are all grouped together. Fourth, the query representation is matched with the music data in the chosen form.

In 2015, the CLAS system used the natural language feature-based parsing facilities in the Python modules distributed as part of the Natural Language ToolKit (NLTK) and a feature-based Context-Free Grammar (CFG) (Wan 2015a, b). The grammar modelled the query as a nested sequence of musical noun phrases. These phrases were 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, solfege nomenclature for notes, and references to scales. One benefit of this approach was that the feature unification facility of the NLTK parsing library could be used to match against feature structures based on the music events.

DMUN (Collins 2014a, b) was the only participant that opted to convert the scores from MusicXML into kern in order to use a large library of sophisticated score analysis tools which this group had previously developed. This approach worked fairly well but there were some anomalies where bar numbering in MusicXML was unorthodox and kern re-numbered it, leading to mis-matches with the gold standard answers. The system split up compound queries such as '… followed by…' to make them comparable to ordinary queries. The MusicXML score was converted to kern and then analysed, leading to several point set representations, each capturing different aspects of the score. Thus, a query component concerning rests could be answered using point sets concerning rests and staff names. Compound queries were dealt with by processing the constituent queries separately and then combining the results.

DMUN's Stravinsqi-Jun2015 algorithm (Katsiavalos and Collins 2015a, b) once again parsed scores in kern format which were converted from MusicXML. With the xml2hum function becoming increasingly out of date, this conversion was problematic for many pieces, which had to be corrected or alternative sources sought. The main differences compared to the 2014 submission were the introduction of a chord-time-intervals function and NLP to split queries by synchronous commands first ('against') and then further by asynchronous commands ('followed by'), such that 'D followed by A against F followed by F' was split first into ('D followed by A' 'F followed by F') and then (('D' 'A') ('F' 'F')). A general question string became the nested list of strings ((s_{1,1} s_{1,2}… s_{1,n(1)}) (s_{2,1} s_{2,2}… s_{1,n(2)})… (s_{m,1} s_{m,2}… s_{m,n(m)})), where each s_{i,j} was a query element (e.g., 'Ab4 eighth note', 'E', 'perfect fifth', 'melodic interval of a 2nd'). Seventeen music-analytic sub-functions were run independently on a given query element. Each function tested whether s_{i,j} was relevant, and, if so, searched for instances of the query in the piece of music, returning a set of time intervals. Subsequent steps of Stravinsqi determined if any combination of these time intervals satisfied the limitations imposed by synchronous and asynchronous question parts.

In 2015, the DMUN system was re-written (Katsiavalos 2016). The group developed a text query parser that, given a sentence such as a C@merata question, generated a script for music operations. The script contained the music concepts and

their relations as described in the query, but in a structured form related to SQL in such a way that workflows of specific music data operations were formed. A parser then read the script and called the corresponding functions from a framework created on top of music21.

KIAM participated for the first time in 2016 (Mytrova 2016). The KIAM system was written in PHP and was based on regular expressions. Queries were both categorised and analysed using these regular expressions; answers were then extracted from raw MusicXML files.

OMDN (Ó Maidín 2014a, b) also used their own software, CPNView, and converted the MusicXML scores into the required format using it. CPNView models a score as an objected-oriented container. This, for example, allows serial access to all staves, serial access to one stave or the use of vertical slices for harmonic analysis. Interestingly, CPNView included an implementation of Alan Forte's system of classification for harmonic analysis (Forte 1973). ONDN used string processing to extract references to notes or rests in the query and to insert the data into a search template. In the case of compound queries, this process repeated for each constituent. The 2015 and 2016 OMDN systems used a similar approach (Ó Maidín 2015a, b, 2016).

TCSL developed a system based on Music21 and the Baseline System (Kini 2014a, b). They took a classical Question Answering approach, defining question classes based on musical features and then defining a search method for each class. They started with a set of nineteen token classes (e.g. a note, a rest, a clef) and searched the query for these, determining for each the value in the query (e.g. the note pitch). A synonym list was developed (e.g. '#'=sharp) to assist with this process. The query was classified into one of fourteen types using tailored rules. Search was limited by an identified scope, e.g. 'on the word' or 'in the treble clef'. Concerning indexing of the score, they worked with Music21 and modelled a score as a sequence of notes annotated with feature information. They also considered indexing by mapping a music feature to a set of passages containing that feature. The 2015 TCSL system was a refinement of this approach (Kini 2015).

The 2014 UNLP system (Asooja et al. 2014a, b) took a query, recognised the musical entities within it, and then searched for them in the corresponding MusicXML file. Six categories of entity were searched for in the query (note, duration, pitch, staff, instrument, clef). Regular expressions were used to recognise these. The MusicXML score was then searched for the entities. The system gave results for single musical entities (such as a note). A compound query combining such entities was handled by searching individually for each component and then combining the results. Two methods of combination were tried in different runs. The 2015 system was a refinement of this approach (Asooja et al. 2015a, b).

5.3 Results

The results for the 3 years can be seen in Tables 18, 19, 20, 21, 22 and 23. As discussed in Sect. 3.4, we developed our own evaluation measures for the task based on Precision, Recall and F-Measure. Precision and Recall are determined in terms of demarcated passages in the music (Sect. 3.3) measured relative to the Gold

Table 18 C@merata 2014 results for all questions

Run	BP	BR	BF	MP	MR	MF
CLAS01	0.713	0.904	0.797	0.764	0.967	0.854
DMUN01	0.372	0.712	0.489	0.409	0.784	0.538
DMUN02	0.380	0.748	0.504	0.417	0.820	0.553
DMUN03	0.440	0.868	0.584	0.462	0.910	0.613
LACG01	0.135	0.101	0.116	0.188	0.142	0.162
OMDN01	0.415	0.150	0.220	0.424	0.154	0.226
TCSL01	0.633	0.821	0.715	0.652	0.845	0.736
UNLP01	0.113	0.516	0.185	0.155	0.703	0.254
UNLP02	0.290	0.512	0.370	0.393	0.692	0.501
Maximum	0.713	0.904	0.797	0.764	0.967	0.854
Minimum	0.113	0.150	0.185	0.155	0.154	0.226
Average	0.420	0.654	0.483	0.460	0.734	0.534

CLAS01 is best run by BF and MF. LACG01 is baseline run—not included in max, min and avg

BP = beat precision, BR = beat recall, BF = beat F-score, MP = measure precision, MR = measure recall, MF = measure F-score (see Sect. 3.4 evaluation)

Table 19 C@merata 2015 results for all questions. CLAS01 is the best run by BF and MF. Results were lower in 2015 than 2014 because the task was harder

Run	BP	BR	BF	MP	MR	MF
CLAS01	0.604	0.636	0.620	0.639	0.673	0.656
DMUN01	0.311	0.739	0.438	0.332	0.788	0.467
DMUN02	0.242	0.739	0.365	0.265	0.809	0.399
DMUN03	0.294	0.739	0.421	0.316	0.794	0.452
OMDN01	0.817	0.175	0.288	0.817	0.175	0.288
TNKG01	0.061	0.488	0.108	0.073	0.586	0.129
UNLP01	0.126	0.430	0.195	0.149	0.508	0.230
Maximum	0.817	0.739	0.620	0.817	0.809	0.656
Minimum	0.061	0.175	0.108	0.073	0.175	0.129
Average	0.351	0.564	0.348	0.370	0.619	0.375

Standards prepared by the organisers. The strict measures (BP, BR, BF) are in terms of the exact start and finish of each passage, measured by beat offset. The lenient measures (MP, MR, MF) only require the start and finish to be in the correct bar (measure).

Results for 2014 are in Table 18. The best run was CLAS01 with BF=0.797 and MF=0.854 (see Table 15 for a runtag listing). These were very high figures which can be attributed both to the high proficiency of CLAS and to the essential simplicity of the questions, 90 of which were concerned with simple notes (see

Table 20 C@merata 2016 results for all questions. DMUN01 is the best run by BF and MF. Note that DMUN01 BP and MP results are respectable at 0.420 and 0.640 but Recall values are very low because of the great complexity of the queries in the 2016 campaign relative to the previous 2 years

Run	BP	BR	BF	MP	MR	MF
DMUN01	0.420	0.038	0.070	0.640	0.058	0.106
KIAM01	0.194	0.011	0.021	0.613	0.035	0.066
OMDN01	0.042	0.004	0.007	0.511	0.044	0.081
UMFC01	0.012	0.038	0.018	0.022	0.073	0.034
Maximum	0.420	0.038	0.070	0.640	0.073	0.106
Minimum	0.012	0.004	0.007	0.022	0.035	0.034
Average	0.167	0.023	0.029	0.447	0.053	0.072

Table 2). TCSL01 also scored very well (BF=0.715, MF=0.736) as did DMUN03 (BF=0.584, MF=0.613). Interestingly, Recall was higher than Precision for all these runs; for example, CLAS01 had BP=0.713, BR=0.904, a difference of 0.191. The difference was particularly large for DMUN03 (BP=0.440, BR=0.868, difference 0.428) so false positive answers were a particular problem for that system. However, part of this effect could have been answers missing from the Gold Standard.

Results for 2015 are in Table 19. Once again, the best run was CLAS01 with BF =0.620 and MF=0.656 (see Table 16 for a runtag listing). Following CLAS01 was DMUN01 (BF=0.438, MF=0.467) while other participants were not close. Note that these figures were much lower than for 2014, mainly because the questions were harder (Table 3). In particular there were follow and synch questions; follow questions allowed a query to specify that one feature (such as a chord) came immediately before another (such as a note). synch questions specified that one feature co-occurred with another. These are musically very interesting and important categories of query but they are clearly much more difficult to answer. During this year, Recall and Precision were similar for CLAS01 (BP=0.604, BR= 0.636, difference 0.032) but not for DMUN01 (BP=0.311, BR=0.739, difference 0.428).

Results for 2016 are in Table 20. CLAS was not able to participate this year, and the best run was DMUN01 with very low figures (BF=0.070, MF=0.106). Questions were very difficult in this year (Tables 4, 5) and substantially more difficult than in 2015. Now the DMUN trend was reversed, with Precision higher than recall (BP=0.420, BR=0.038, difference 0.382, MP=0.640, MR=0.058, difference 0.582). However this was in fact a different system from the previous year.

Over the 3 years, what was the relative difficulty of the questions? Looking at Table 21 for 2014 (with query types in Table 2), we can use the BF score as a measure. simple_length (BF=0.810), simple_pitch (BF=0.677) and pitch_and_length (BF=0.644) were the easiest; this is not surprising as such queries are very straightforward. After that we have word_spec (BF=0.520), stave_spec (BF= 0.508), melodic_interval (BF=0.402), perf_spec (BF=0.339) and followed_by (BF

Table 21 C@merata 2014 average results by question type

Type	BP	BR	BF	MP	MR	MF
simple_pitch	0.645	0.736	0.677	0.685	0.787	0.720
simple_length	0.780	0.846	0.810	0.830	0.906	0.864
pitch_and_length	0.662	0.726	0.644	0.719	0.803	0.710
perf_spec	0.339	0.547	0.339	0.350	0.582	0.352
stave_spec	0.408	0.682	0.508	0.432	0.732	0.540
word_spec	0.487	0.771	0.520	0.487	0.771	0.520
followed_by	0.291	0.518	0.278	0.351	0.716	0.355
melodic_interval	0.396	0.417	0.402	0.471	0.501	0.481
harmonic_interval	0.185	0.207	0.188	0.269	0.329	0.281
cadence_spec	0.071	0.141	0.093	0.171	0.297	0.214
triad_spec	0.081	0.125	0.095	0.124	0.171	0.138
texture_spec	0.060	0.109	0.075	0.072	0.141	0.092

Refer to Table 2 for example queries of these types

Table 22 C@merata 2015 average results by question type

Type	BP	BR	BF	MP	MR	MF
1_melod	0.450	0.764	0.508	0.467	0.801	0.531
n_melod	0.216	0.378	0.249	0.236	0.472	0.276
1_harm	0.261	0.426	0.285	0.289	0.471	0.317
texture	0.000	0.000	0.000	0.143	0.061	0.086
follow	0.172	0.415	0.217	0.247	0.486	0.275
synch	0.193	0.373	0.178	0.235	0.425	0.208
perf qualified	0.359	0.552	0.230	0.362	0.578	0.236
instr qualified	0.426	0.488	0.308	0.440	0.522	0.326
clef qualified	0.329	0.588	0.342	0.339	0.615	0.355
time qualified	0.187	0.476	0.248	0.211	0.544	0.281
key qualified	0.143	0.089	0.110	0.291	0.357	0.300

Refer to Table 3 for example queries of these types

=0.278). Here, we are linking two pieces of information together; for example in stave_spec we are specifying the note 'D4…' and the stave '…in the right hand'. After this group of query types, there is a big gap to harmonic_interval (BF=0.188). Harmonic intervals can be across staves and thus harder to spot, especially as the notes in question are not required to be the same length in our task. Finally, we have triad_spec (BF=0.095), cadence_spec (0.093) and texture_spec (0.075). These are clearly hard features to find.

Turning to Table 22 for 2015 (with query types in Table 3), 1_melod (BF=0.508) are the equivalent of simple_pitch, simple_length and pitch_and_length in the previous year and hence are the easiest to answer. After that, and substantially behind, we have 1_harm (BF=0.285), n_melod (BF=0.249), follow (BF=0.217)

Table 23 C@merata 2016 average results by question type

Type	BP	BR	BF	MP	MR	MF
1_melod	0.232	0.044	0.054	0.520	0.101	0.129
n_melod	0.125	0.016	0.028	0.384	0.051	0.086
1_harm	0.076	0.023	0.019	0.300	0.033	0.035
n_harm	0.063	0.007	0.013	0.128	0.032	0.030
texture	0.000	0.000	0.000	0.000	0.000	0.000
follow	0.317	0.047	0.078	0.458	0.076	0.126
synch	0.000	0.000	0.000	0.103	0.011	0.018

Refer to Tables 4 and 5 for example queries of these types. Note that in 2016 follow and synch questions were across 1_melod, n_melod, 1_harm and n_harm

and synch (BF=0.178). These are all quite difficult; 1_harm is concerned with individual chords across staves, n_melod requires searching for melodic patterns, some of which were complex (see Table 3); follow and synch are the most difficult because they relate two independent features both of which may be hard to spot even independently. The texture query type scored BF=0.000, so no system could answer any of these. Textures are somewhat difficult to demarcate precisely in terms of where they start and finish, but any musician can spot a feature like 'Alberti bass' or 'homophony' without difficulty. The last five rows of Table 22 give the scores for queries—across the other categories—which were qualified in different ways. All were intermediate in difficulty and are in the order clef qualified e.g. 'in the bass clef' (BF=0.342), instr(ument)_qualified e.g. 'on the oboe' (BF=0.308), time (signature) qualified (BF=0.248) e.g. 'in 4/4' perf(ormance) qualified e.g. 'trill on a' (BF=0.230) and key qualified e.g. 'in F major' (BF=0.110). Clearly these types of modification are of considerable use in narrowing down the set of passages which will match a query and are well worth having in a system.

Now, turning to Table 23 for 2016 (with query types in Tables 4 and 5), the figures are all very low, so not much can be concluded from them. 1_melod (BF=0.054) were the easiest, and after that n_melod (BF=0.028), 1_harm (BF=0.019), n_harm (BF=0.013) and texture (BF=0.000). Of the qualifications over the other categories of query, follow (BF=0.078) were easier than synch (BF=0.000).

Finally, what about the difference between BF and MF scores? In C@merata we devised strict and lenient measures; the strict measure insisted that the exact beat in the bar was correct for both the start and end of the feature; the lenient measure, on the other hand, only required the correct bar to be specified for the start and end (features could span more than one bar). We had expected that the scores for the lenient measure would be much higher than those for the strict measure, but this was not the case. Returning to Tables 18, 19 and 20 and considering the top scoring run we have 2014 (Table 18) CLAS01 (BF=0.797, MF=0.854, difference 0.057); 2015 (Table 19) CLAS01 (BF=0.620, MF=0.656, difference 0.036); 2016 (Table 20) DMUN01 (BF=0.070, MF=0.106, difference 0.036). All these differences are quite small, suggesting that, if you know the bar(s) where a passage starts and ends, it is not much more difficult to find the exact beat in the bar in each case.

6 C@merata and musicology

The long-term aim of C@merata is to develop the technology to analyse linguistic expressions in real musicological texts and to link these to scores. In a recent study (Sutcliffe et al. 2015a) we analysed some actual published texts to judge how close we were to achieving our goal. We first chose three important text sources. The first was an analysis of the Beethoven Symphonies by Antony Hopkins (Chapter 2: Symphony No. 1 in C Major Op. 21) (Hopkins 1982) (henceforth ah). The second was the study of Domenico Scarlatti by Ralph Kirkpatrick (Chapter 10: Scarlatti's Harmony, Section Cadential vs. Diatonic Movement of Harmony) (Kirkpatrick 1953) (henceforth rk). The third was the entry for Anton Bruckner by Deryck Cooke (Section 7. Music) (Cooke 1995) from the New Grove Dictionary of Music and Musicians (Sadie 1980) (henceforth dc).

We extracted phrases from the above works by hand—261 in all—and organised them into fourteen categories: notes, intervals, scales, melodies, rhythms, tempi, dynamics, keys, harmony, counterpoint, texture & instrumentation, bar numbers, passages & sections and structures & sequences. Furthermore, they are classed as Specific or Vague. Examples of each category can be seen in Table 24, with two Specific and two Vague for each phrase type. The source is indicated in square brackets: [ah26] means ah (i.e. Hopkins) p26; [dc364lh] means dc (i.e. Cooke) p364 in Grove, left hand column. It is important to note that the categories in Table 24 are for illustration only and are neither exhaustive nor mutually exclusive. The examples are given purely to illustrate the kinds of references to musical passages which one might find in a musicological text. Moreover, the binary categorisation into Specific and Vague was also purely for illustration purposes as specificity lies on a scale. However, this study did allow us to draw some interesting conclusions of relevance to C@merata.

The first point to note is that the references vary in specificity; some are clear and unambiguous (C#-D rising semitone, D major, eight-part choir, bars 189–198); others are much more difficult to pin down (alien F#, disturbing syncopations, anguished D minor chromaticism, varied alternation of two long-drawn themes). In our task we have both specific (e.g. 'dotted quarter note' from 2014) and less specific (e.g. 'five note melody in bars 1–10' from 2015) types of queries.

Secondly, however, all the phrases are meaningful—an expert familiar with the works concerned is likely to be able to identify the points mentioned in the score with a fair accuracy (high Precision even if not necessarily high Recall). This suggests that they are interesting and worthwhile to study.

Thirdly, some categories of phrase in the real examples of Table 24 are both simple and clear. Examples include Notes ('G'), Intervals ('ascending diminished fifth'), Scales ('D major'), Rhythms ('repeated crotchet chords'), Dynamics ('FF'), Keys ('Bb minor') and bar/measure numbers ('measure thirteen'). We have such queries in our task, and our participants have been successful at matching them to scores.

Fourthly and conversely however, some categories of phrase can be both complex and imprecise. Examples include Texture and Instrumentation ('a faint

Table 24 Fourteen types of referring expressions taken from actual musicological texts

Category	S/V	Examples
Notes	S	[ah26] giant unison G from the entire orchestra
		[rk220] based on nothing else but A, D, E, and A
	V	[ah12] alien F# in the ascending scale
		[dc364lh] pedal point
Intervals	S	[ah24] C#-D rising semitone
		[dc363lh] an ascending diminished fifth
	V	[ah19] fragment of five rising crotchets
		[dc364lh] themes based on falling octaves
Scales	S	[dc363lh] parts entering successively on the degrees of the ascending scale of D major
		[dc363rh] old church modes… Phrygian and Lydian
	V	[ah28] the initial scale
		[ah29] little scales dart to and fro
Melodies	S	[ah13] semiquaver descent in bar 18
		[ah19] fragment of five rising crotchets
	V	[ah19] Second Subject appearing in the tonic key
		[dc363rh] the chorale themes in the symphonies
Rhythms	S	[ah15] quaver pattern
		[ah25] repeated crotchet chords
	V	[ah18] disturbing syncopations
		[dc364lh] hammering ostinatos
Tempi	S	[ah11] slow tempo
		[dc366lh] slow movements
	V	[ah28] rustic oom-pah bass
		[dc364rh] intense and long-drawn string cantabile
Dynamics	S	[ah26] violins in bar 126 come in FF
		[ah29] sudden fortissimo outburst
	V	[ah29] sudden roaring
		[dc364lh] murmering tremolando
Keys	S	[ah10] D major
		[dc366lh] in Bb minor
	V	[rk221] modulatory excursion of the second half
		[dc363rh] unusual key changes
Harmony	S	[rk221] major dominant
		[dc364lh] tonic triad of E major
	V	[rk220] departure from three-chord harmony
		[dc363lh] anguished D minor chromaticism
Counterpoint	S	[ah23] cellos provide a delicate countertune
		[dc363lh] parts entering successively on the degrees of the ascending scale of D major
	V	[rk220] dominated by diatonic movement of parts
		[dc363lh] bold polyphonic imitation of a single point

Table 24 continued

Category	S/V	Examples
Texture, Instrumentation	S	[dc363lh] eight-part choir
		[dc363rh] a piece of unison plainsong
	V	[ah29] decked with garlands of scales from flutes, clarinets and bassoons
		[dc364rh] a faint background sound, emerging almost imperceptibly out of silence
Bar numbers	S	[ah15] bars 189–198
		[rk220] measure thirteen to measure fifteen
	V	[ah24] sixteen or at most thirty-two bars long
		[dc365rh] over periods of 16, 32 or even 64 bars
Passages, Sections	S	[dc363rh] whose slow movement and finale
		[dc364rh] far-ranging first movement
	V	[rk220] series of small sequential passages
		[dc362rh] a passage from the Gloria
Structures, Sequences	S	[ah18] First Subject
		[rk221] Phrygian cadence
	V	[dc365rh] exposition (nearly always built on three subject groups rather than two)
		[dc366rh] varied alternation of two long-drawn themes

They are categorised into Specific (S) and Vague (V). The source is indicated in square brackets: [ah26] means ah (i.e. Hopkins) p26; [dc364lh] means dc (i.e. Cooke) p364 in Grove, left hand column

background sound, emerging almost imperceptibly out of silence'), Passages and Sections ('a passage from the Gloria') and Structures and Sequences ('exposition (nearly always built on three subject groups rather than two)'). Western classical music excels in structure and in harmony, so treatment of these topics tends to be particularly interesting and important. The richness and ambiguity of language are its strengths in this context as a great deal can be suggested in relatively few words. Moreover, to the expert, the references remain quite clear, though a considerable amount of knowledge and background information is being brought to bear. We are only starting to approach such queries in the task. For example, in 2016 we had 'imitative texture in bars 1–18' and 'counterpoint in bars 1–14', both of which are vague. However, at present, while we can recognise instances of such queries (i.e. we have an idea what to look for) we do not at present have the means to find matches when so little is actually specified and so much depends on a deep musical knowledge.

Fifthly, it is interesting to observe that many of the examples in Table 1 are noun phrases; this construct can express very complicated and detailed concepts in a musicological text and it is actually used; so this validates an evaluation task which specialises in noun phrases.

Sixthly, phrases in natural language can never be replaced by expressions in a pattern language (such as regular expressions applied over text strings). Such

expressions are by their nature unambiguous and in practical contexts they are usually concise. Therefore, the study of natural language in musicology is not made unnecessary by the existence of such languages. On the other hand, such expression languages are extremely useful and worthwhile (Viglianti 2015); we are working on them as part of the 2017 task. One possible application of them here is to map a natural language phrase onto a pattern (possibly extremely complex) in such an expression language in order to initiate a search.

A final point to make here is that the interestingness of a particular type of query depends on the context and application. In Table 24 we are looking at static texts. What about dynamic dialogues? We believe that noun phrase queries could have great value there as well. Some of our queries seem quite straightforward, e.g. 'two minims followed by a crotchet rest'. However, in a search context, the ability to submit such queries could be extremely useful as could the ability to vary the specificity (e.g. 'two notes followed by a rest'). Natural language is extremely good for this.

7 Summary and conclusions

C@merata is an evaluation task where the inputs are firstly a noun phrase describing a musical feature, and secondly a symbolic music score in MusicXML. The required output is a list of one or more matching passages in the score. Each passage has a start and end, specified in terms of bar (measure) and beat, using the *divisions* concept of MusicXML to cope with any length of time in any time signature, including n-tuplets. So far there have been three annual campaigns, each with twenty scores and 200 questions. What has this work shown and how successful was it?

NLP has been used for musical analysis for a considerable period (see Sect. 2): The text of song lyrics was analysed as early as 2003 by Brochu and de Freitas, using IR techniques, with Mahedero et al. (2005) following on with some NLP approaches. However, QA against text specifically on musical topics did not exist prior to the QA4MRE evaluations which we started in 2011 (see Table 1, Sect. 2). Moreover, we believe the C@merata evaluations are the first to combine QA with MIR in a detailed and systematic way, working with symbolic scores, and embodying a deep knowledge of music theory.

Within C@merata, we have developed a paradigm in which we can express a query and evaluate the answers using versions of Precision and Recall, in both Strict and Lenient forms (see Sect. 3.4). As we saw, Beat Precision (strict) requires the start and end of a passage to be at exactly the right beat, while Measure Precision (lenient) only requires the start and end to lie in the correct measure (bar). This has worked very well. The use of a 'vertical line' through a score, not specifying which staffs are involved, is a simplification but it results in a very workable task. Moreover, most musicologists, given the bar (measure) can find the required feature at a glance; the problem for them is that there are hundreds of bars in a large score and, of course, an unlimited number of scores.

Concerning Strict and Lenient, our results show that there is surprisingly little difference between them; in 2014 (Table 18) average BF was 0.483 and average MF was 0.534. In the 2015 (Table 19) the figures were BF=0.348 and MF=0.375. This suggests that if you can find the bar, you can find the exact passage in most cases.

How realistic and useful were the queries? In 2014 we started off with some very basic ones, many looking for simple notes (e.g. 'D# crotchet', Table 2). By 2016 we had added far more complex queries, some very specific and long (e.g. 'crotchet, crotchet rest, crotchet rest, crotchet, crotchet rest, crotchet, crotchet, crotchet, crotchet, crotchet in the Timpani', Table 3) and some quite vague ('all three violin parts in unison in measures 1–59', Table 4). Moreover, we started off by creating our queries according to a 'template' manifested in the detailed Task Description but then started deriving them from real sources including music exam papers.

How good were participants at performing the task? Generally, very few people have a detailed knowledge of classical music theory and natural language processing. Morever, a knowledge of music information retrieval work is required, going back to Downie's ground-breaking experiments using note n-grams (Downie and Nelson 2000). There are very few such people, and our participants have tended to be more expert in MIR than NLP. In the early years, they still achieved remarkable successes, especially CLAS and DMUN. However, grammatical analysis of the queries has become quite complicated because of the multiplicity of complex structured musical terminology. Participants struggled with this though we have since produced an accurate analyser ourselves (Sutcliffe and Liem 2017; Sutcliffe et al. 2017). We plan to make the output of this available to participants in future editions of the C@merata task, in order to facilitate the initial processing.

What does this work reveal about how NLP relates to MIR? First, our studies have shown that natural language can be used to make extremely complex and detailed references to musical events which can potentially be retrieved by MIR. Such references can be found in actual musicological texts and these can be studied further within a C@merata paradigm where queries are created manually for an evaluation. Second, because of the subtlety of language, natural language text can express more than strings in a formal language (e.g. Regular Expressions). For example, language can be very vague or very specific. Moreover, language can be subjective as well as objective. Thus, our work can tell us something about language itself which has not previously been investigated, as well as how language relates to music. Third, both we and our participants have demonstrated some initial methods by which a natural language text can be converted into an MIR query. However, much remains to be done.

We conclude with a mention of Downie's (2003) statement of seven Facets of Music: Pitch, Temporal, Harmonic, Timbral, Editorial, Textual and Bibliographic. In a natural language context, we have made substantial progress in describing and searching within scores for references in natural language to Pitch, Temporal and Harmonic features in a Textual context. Timbral features have only begun to be addressed, and we have not really looked at Editorial or Bibliographic aspects, though they are both related to some of the work we have done.

References

Artstein, R., & Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics, 34*(4), 555–596.

Asooja, K., Ernala, S. K., & Buitelaar, P. (2014a). UNLP at the MediaEval 2014 C@merata task. In *Proceedings of the MediaEval 2014 workshop Barcelona, Catalunya, Spain, October 16–17, 2014*. http://ceur-ws.org/Vol-1263/mediaeval2014_submission_52.pdf. Accessed 15 Mar 2017.

Asooja, K., Ernala, S. K., & Buitelaar, P. (2014b). UNLP at the C@merata task: Question answering on musical scores. In *Proceedings of the C@merata task at MediaEval 2015*. http://csee.essex.ac.uk/camerata/unlp_asooja_long_14.pdf. Accessed 15 Mar 2017.

Asooja, K., Ernala, S. K., & Buitelaar, P. (2015a). UNLP at the MediaEval 2015 C@merata task. In *Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/Paper86.pdf. Accessed 15 Mar 2017.

Asooja, K., Ernala, S. K., & Buitelaar, P. (2015b). UNLP at the 2015 C@merata task: question answering on musical scores by matching the passage sequences to musical entity sequences. http://csee.essex.ac.uk/camerata/unlp_asooja_long_15.pdf. Accessed 15 Mar 2017.

Baumann, S. (2003). Cultural metadata for artist recommendation. In *Proceedings of the WEDELMUSIC*.

Brochu, E., & de Freitas, N. (2003). "Name that song!": A probabilistic approach to querying music and text. In: S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15, pp. 1505–1512). Cambridge, MA: MIT Press.

Buffa, M., & Cabrio, E. (2016). Natural language processing of song lyrics—WASABI project (web audio semantic aggregated in the browser for indexation). http://wimmics.inria.fr/node/60, http://wasabihome.i3s.unice.fr/. Accessed 22 Mar 2017.

Cleverdon, C. W. (1962). *Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems*. Cranfield, England: College of Aeronautics.

Collins, T. (2014a). Stravinsqi/De Montfort University at the MediaEval 2014 C@merata task. In *Proceedings of the MediaEval 2014 Workshop Barcelona, Catalunya, Spain, October 16–17, 2014*. http://ceur-ws.org/Vol-1263/mediaeval2014_submission_50.pdf. Accessed 29 Mar 2017.

Collins, T. (2014b). Stravinsqi/De Montfort University at the MediaEval 2014 C@merata task. In *Proceedings of the C@merata Task at MediaEval 2015*. http://csee.essex.ac.uk/camerata/dmun_collins_long_14.pdf.

Cooke, D. (1995). Bruckner, (Joseph) Anton. In S. Sadie (Ed.), *New Grove dictionary of music and musicians* (Vol. 3, Section 7, pp. 362–366). London, UK: Macmillan.

Cuthbert, M. S., & Ariza C. (2010). music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the international symposium on music information retrieval, Utrecht, The Netherlands, August 09–13, 2010* (pp. 637–642).

Downie, J. S. (2003). Music information retrieval. *Annual Review of Information Science and Technology, 37,* 295–340.

Downie, J. S. (2008). The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology, 29*(4), 247–255. https://doi.org/10.1250/ast.29.247.

Downie, S., & Nelson, M. (2000). Evaluation of a simple and effective music information retrieval method. In *Proceedings of the ACM international conference on research and development in information retrieval* (SIGIR) (pp. 73–80).

Dunning, T. (1993). Statistical identification of language. *ACM Transactions on Programming Languages and Systems, 15*(5), 745–770.

Ekman, P. (1993). Facial expression of emotion. *American Psychologist, 48*, 384–392.

Forte, A. (1973). *The structure of atonal music*. New Haven: Yale University Press.

Fux, J. J. (1725). *Gradus ad Parnassum* (practical rules for learning composition translated from a work intitled Gradus ad Parnassum written originally in Latin by John Joseph Feux). Translated around 1750 by unknown translator. London: Welcker. http://imslp.nl/imglnks/usimg/3/31/IMSLP370587-PMLP187246-practicalrules fo00fuxj.pdf.

Gravier, G., Demarty, C. -H., Bredin, H., Ionescu, B., Boididou, C., Dellandrea, E., Choi, J., Riegler, M., Sutcliffe, R., Szoke, I., Jones, G., & Larson, M. (2016). *Proceedings of the MediaEval 2016 Workshop, Hilversum, The Netherlands, October 20–21 2016*. http://ceur-ws.org/Vol-1739/. Accessed 1 Apr 2017.

Hopkins, A. (1982). *The nine symphonies of Beethoven*. London: Pan Books.

Hu, X., Choi, K., Hao, Y., Cunningham, J., Lee, J. H., Laplante, A., Bainbridge, D., & Downie, J. S. (2017). Exploring the music library association mailing list: A text mining approach. In *Proceedings of ISMIR, Suzhou, China, 2017.*

Hu, X., Downie, J. S., & Ehman, A. F. (2009). Lyric text mining in music mood classification. In *Proceedings of ISMIR* (pp. 411–416).

Huron, D. (1997). Humdrum and Kern: Selective feature encoding. In E. Selfridge-Field (Ed.), *Beyond MIDI* (pp. 375–401). Cambridge, MA: MIT Press.

Huron, D. (2002). Music information processing using the Humdrum toolkit: Concepts, examples, and lessons. *Computer Music Journal, 26*(2), 11–26.

Katsiavalos, A., (2016). DMUN: A textual interface for content-based music information retrieval in the C@merata task for MediaEval 2016. In *Proceedings of the MediaEval 2016 workshop, Hilversum, The Netherlands, October 20–21, 2016*. http://ceur-ws.org/Vol-1739/MediaEval_2016_paper_56.pdf . Accessed 26 Mar 2017.

Katsiavalos, A., & Collins, T. (2015a). DMUN at the MediaEval 2015 C@merata task: The Stravinsqi algorithm. In *Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/Paper88.pdf. Accessed 29 Mar 2017.

Katsiavalos, A., & Collins, T. (2015b). DMUN at the C@merata 2015 task: A full description of the Stravinsqi-June2015 algorithm. In *Proceedings of the C@merata task at MediaEval 2014*. http://csee.essex.ac.uk/camerata/dmun_collins_long_15.pdf. Accessed 29 Mar 2017.

Kini, N. (2014a). TCSL at the MediaEval 2014 C@merata task. In *Proceedings of the MediaEval 2014 workshop Barcelona, Catalunya, Spain, October 16–17, 2014*. http://ceur-ws.org/Vol-1263/mediaeval2014_submission_51.pdf. Accessed 29 Mar 2017.

Kini, N. (2014b). TCSL at the C@merata 2014 task: A tokenizing and parsing framework to understand queries on sheet music. In *Proceedings of the C@merata task at MediaEval 2015.* http://csee.essex.ac.uk/camerata/tcsl_kini_long_14.pdf. Accessed 29 Mar 2017.

Kini, N. (2015). TNKG at the MediaEval 2015 C@merata task. In *Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/Paper89.pdf. Accessed 29 Mar 2017.

Kirkpatrick, R. (1953). *Domenico Scarlatti*. Princeton, NJ: Princeton University Press.

Kitson, C. H. (1907). *The art of counterpoint and its application as a decorative principle*. Oxford, UK: Clarendon Press. https://archive.org/details/artofcounterpoin00kitsuoft. Accessed 1 Jan 2017.

Kuribayashi, T., Asano, Y., & Yoshikawa, M. (2013). Ranking method specialized for content descriptions of classical music. In *Proceedings of the 22nd international world wide web conference, Rio de Janeiro, Brazil, 13–17 May, 2013* (pp. 141–142). http://dl.acm.org/citation.cfm?id=2487856. Accessed 1 Apr 2017.

Kuribayashi, T., Asano, Y., & Yoshikawa, M. (2015). Towards support for understanding classical music: Alignment of content descriptions on the web. In *Proceedings of ISMIR 2015*. http://ismir2015.uma.es/articles/287_Paper.pdf. Accessed 1 Apr 2017.

Larson, M., Ionescu, B., Anguera, X., Eskevich, M., Korshunov, P., Schedl, Soleymani, M., Petkos, G., Sutcliffe, R., Choi, J., & Jones, G. F. (2014). *Proceedings of the MediaEval 2014 workshop, Barcelona, Spain, October 16–17 2014*. http://ceur-ws.org/Vol-1263/. Accessed 1 Mar 2017.

Larson, M., Ionescu, B., Sjöberg, M., Anguera, X., Poignant, J., Riegler, M., Eskevich, M., Hauff, C., Sutcliffe, R., Jones, G. F., Yang, Y. -H., Soleymani, M., & Papadopoulos, S. (2015). *Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/. Accessed 1 Mar 2017.

Lavecchia, C., Smaïli, K., & Langlois, D. (2007). Building parallel corpora from movies. In *The 4th international workshop on natural language processing and cognitive science - NLPCS 2007, Jun 2007*. Funchal, Madeira.

Logan, B., Kositsky, A., & Moreno, P. (2004). Semantic analysis of song lyrics. In *IEEE ICME*.

Mahedero, J. P. G., MartÍnez, Á., Cano, P., Koppenberger, M., & Gouyon, F. (2005). Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on multimedia (MULTIMEDIA 2005)* (pp. 475–478). New York, NY, USA: ACM.

McKay, C., et al. (2010). Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. In: *Proceedings of ISMIR*.

Mihalcea, R., & Strapparava, C. (2012). Lyrics, music, and emotions. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural*

language learning (EMNLP-CoNLL 2012) (pp. 590–599). Stroudsburg, PA, USA: Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the international conference on learning representations, ICLR, Scottsdale, AZ, 2013*. arXiv:1301.3781.

Mollá, D., & Vicedo, J. L. (2007). Question answering in restricted domains: An overview. *Computational Linguistics, 33*(1), 41–61.

Mytrova, M. (2016). The KIAM system in the C@merata task at MediaEval 2016. In *Proceedings of the MediaEval 2016 workshop, Hilversum, The Netherlands, October 20–21, 2016*. http://ceur-ws.org/Vol-1739/MediaEval_2016_paper_57.pdf. Accessed 14 Mar 2017.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes, 30*(1), 3–26.

Ó Maidín, D. S. (2014a). OMDN at the MediaEval 2014 C@merata task. In *Proceedings of the MediaEval 2014 workshop Barcelona, Catalunya, Spain, October 16–17, 2014*. http://ceur-ws.org/Vol-1263/mediaeval2014_submission_48.pdf. Accessed 12 Mar 2017.

Ó Maidín, D. S. (2014b). OMDN at the MediaEval 2014 C@merata task: Using CPNView to answer questions about scores. In *Proceedings of the C@merata task at MediaEval 2015*. http://csee.essex.ac.uk/camerata/omdn_omaidin_long_14.pdf. Accessed 12 Mar 2017.

Ó Maidín, D. S. (2015a). OMDN at the MediaEval 2015 C@merata task. In *Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/Paper88.pdf . Accessed 12 Mar 2017.

Ó Maidín, D. S. (2015b). OMDN at the C@merata 2015 task: A description of the CPNView approach to answering natural language questions about music scores. In *Proceedings of the C@merata task at MediaEval 2014.* http://csee.essex.ac.uk/camerata/omdn_omaidin_long_15.pdf. Accessed 12 Mar 2017.

Ó Maidín, D. (2016). OMDN at the C@merata 2016 task: Approaches to and issues arising from answering natural language questions about music scores. In *Proceedings of the MediaEval 2016 workshop, Hilversum, The Netherlands, October 20–21, 2016.* http://ceur-ws.org/Vol-1739/MediaEval_2016_paper_58.pdf. Accessed 16 Mar 2017.

O'Hara, T. (2011). Inferring the meaning of chord sequences via lyrics. In *Proceedings of 2nd workshop on music recommendation and discovery (WOMRAD 2011), Chicago, IL, October.*

Oramas, S. (2017). Knowledge extraction and representation learning for music recommendation and classification. Ph.D. Thesis, Universitat Pompeu Fabra Barcelona, 14th November, 2017. https://zenodo.org/record/1100973. Accessed 1 Mar 2018.

Oramas, S., Espinosa-Anke, L., Sordo, M., Saggion, H., & Serra, X. (2016a). Information extraction for knowledge base construction in the music domain. *Journal of Knowledge & Data Engineering, 106*, 70–83.

Oramas, S., Espinosa-Anke, L., Saggion, H., & Serra, X. (2016b). ELMD: An automatically generated entity linking gold standard dataset in the music domain. In *Proceedings of LREC 2016.*

Oramas, S., Espinosa-Anke, L., Lawlor, A., Serra, X., & Saggion, H. (2016c). Exploring music reviews for music genre classification and evolutionary studies. In *17th international society for music information retrieval conference. ISMIR 2016.*

Oramas, S., Espinosa-Anke, L., Zhang, S., Saggion, H., & Serra, X. (2016d). Natural language processing for music information retrieval. *Tutorial held at 17th international society for music information retrieval conference (ISMIR 2016).* http://mtg.upf.edu/system/files/Tutorial_NLP4MIR.pdf. Accessed 1 September 2016.

Oramas, S., Espinosa-Anke, L., Saggion, H., & Serra, X. (2017a). Natural language processing for music information retrieval. *Tutorial, Universitat Pompeu Fabra, Barcelona, 30th January 2017*. http://mtg.upf.edu/system/files/Tutorial_NLP4MIR_2017.pdf. Accessed 1 Feb 2017.

Oramas, S., Nieto, O., Barbieri, F., & Serra, X. (2017b). Multi-label music genre classification from audio, text, and images using deep features. In *Proceedings of the 18th international society of music information retrieval conference ISMIR 2017.*

Oramas, S., & Sordo, M. (2016). Knowledge is out there: A new step in the evolution of music digital libraries. *Fontes Artis Musicae, 63*(4), 285–298.

Oramas, S., Sordo, M., Espinosa-Anke, L., & Serra, X. (2015). A semantic-based approach for artist similarity. In *16th international society for music information retrieval conference (ISMIR 2015).*

Oramas, S., Sordo, M., & Serra, X. (2014). Automatic creation of knowledge graphs from digital musical document libraries. In *Conference in Interdisciplinary Musicology (CIM 2014)*.

Orio, N. (2006). Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval, 1*(1), 1–90.

Peñas, A., Forner, P., Rodrigo, A., Sutcliffe, R., Forăscu, C., Mota, C. (2010). Overview of ResPubliQA 2010: Question answering evaluation over European Legislation. In *Proceedings of ResPubliQA 2010. Held as part of CLEF 2010*. http://csee.essex.ac.uk/staff/rsutcl/ResPubliQA2010_Overview-final.pdf. Accessed 10 Mar 2017.

Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, A., Forascu, C., Alegria, I., Giampiccolo, D., Moreau, N., & Osenova, P. (2009). Overview of ResPubliQA 2009: Question answering evaluation over European legislation. In *Notebook of the cross language evaluation forum, CLEF 2009, Corfu, Greece, 30 September–2 October*. http://csee.essex.ac.uk/staff/rsutcl/CLEF2009wn-QACLEF-PenasEt2009.pdf. Accessed 10 Mar 2017.

Peñas, A., Hovy, E., Forner, P., Rodrigo, A., Sutcliffe, R., Forascu, C., & Sporleder, C. (2011). Overview of QA4MRE at CLEF 2011: question answering for machine reading evaluation. In *Proceedings of QA4MRE-2011. Held as part of CLEF 2011.* http://csee.essex.ac.uk/staff/rsutcl/QA4MRE-2011_Overview-final.doc. Accessed 10 Mar 2017.

Peñas, A., Hovy, E., Forner, P., Rodrigo, Á., Sutcliffe, R., & Morante, R. (2013). QA4MRE 2011–2013: Overview of question answering for machine reading evaluation. In P. Forner, H. Müller, R. Paredes, P. Rosso, & B. Stein (Eds.), *Information access evaluation. Multilinguality, multimodality, and visualization. CLEF 2013.* (Vol. 8138)., Lecture notes in computer science Berlin: Springer. https://doi.org/10.1007/978-3-642-40802-1_29.

Peñas, A., Hovy, E., Forner, P., Rodrigo, A., Sutcliffe, R., Sporleder, C., Forascu, C., Benajiba, Y., & Osenova, P. (2012a). Overview of QA4MRE at CLEF 2012: Question answering for machine reading evaluation. In *Proceedings of QA4MRE-2012. Held as part of CLEF 2012*. http://csee.essex.ac.uk/staff/rsutcl/QA4MRE-2012-overview-v11.pdf.

Peñas, A., Magnini, B., Forner, P., Sutcliffe, R., Rodrigo, A., & Giampiccolo, D. (2012b). Question answering at the cross-language evaluation forum 2003–2010. *Language Resources and Evaluation Journal, 46*(2), 177–217.

Peñas, A., & Rodrigo, A. (2011). A simple measure to assess non-response. In *Proceedings of 49th annual meeting of the association for computational linguistics: Human language technologies (ACL-HLT 2011)*, *Portland, Oregon, USA. June 19–24. 2011.*

Read, G. (1978). *Music notation—A manual of modern practice*. London, UK: Victor Gollancz.

Sadie, S. (Ed.). (1980). *The new Grove dictionary of music and musicians*. London, UK: Macmillan.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Schedl, M., Gómez, E., & Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval, 8*(2–3), 127–261.

Socher, R., Bauer, J., Manning, C. D., & Ng, A. Y. (2013). Parsing with compositional vector grammars. In *Proceedings of ACL 2013.*

Sterckx, L., Demeester, T., Deleu, J., Mertens, L., & Develder, C. (2014). Assessing quality of unsupervised topics in song lyrics. In *36th European Conference on Information Retrieval, Lecture Notes in Computer Science (ECIR 2014).*

Sutcliffe, R. F. E. (2014). *A description of the C@merata baseline system in Python 2.7 for answering natural language queries on MusicXML scores*. University of Essex Technical Report, 21st May, 2014.

Sutcliffe, R. F. E., Collins, T., Hovy, E., Lewis, R., Fox, C., & Root, D. L. (2016). The C@merata task at MediaEval 2016: Natural language queries derived from exam papers, articles and other sources against classical music scores in MusicXML. In *Proceedings of the MediaEval 2016 workshop, Hilversum, The Netherlands, October 20–21, 2016*. http://ceur-ws.org/Vol-1739/MediaEval_2016_paper_55.pdf. Accessed 4 Feb 2017.

Sutcliffe, R. F. E., Crawford, T., Fox, C., Root, D. L., & Hovy, E. (2014a). The C@merata task at MediaEval 2014: Natural language queries on classical music scores. In *Proceedings of MediaEval 2014 workshop, Barcelona, Spain, October 16–17 2014*. http://ceur-ws.org/Vol-1263/mediaeval2014_submission_46.pdf. Accessed 12 Mar 2017.

Sutcliffe, R. F. E., Crawford, T., Fox, C., Root, D. L., & Hovy, E. (2014b). Shared evaluation of natural language queries against classical music scores: A full description of the C@merata 2014 task. In

Proceedings of the C@merata task at MediaEval 2014. http://csee.essex.ac.uk/camerata/newest_camerata_long_14.pdf. Accessed 12 Mar 2017.

Sutcliffe, R. F. E., Crawford, T., Fox, C., Root, D. L., Hovy, E., & Lewis, R. (2015a). Relating natural language text to musical passages. In *Proceedings of the 16th international society for music information retrieval conference, Malaga, Spain, 26–30 October, 2015*. http://ismir2015.uma.es/articles/263_Paper.pdf. Accessed 20 Mar 2017.

Sutcliffe, R. F. E., Fox, C., Root, D. L., Hovy, E., & Lewis, R. (2015b). Shared evaluation of natural language queries against classical music scores: A full description of the C@merata 2015 task. In *Proceedings of the C@merata task at MediaEval 2015*. http://csee.essex.ac.uk/camerata/camerata_long_15.pdf. Accessed 4 Mar 2017.

Sutcliffe, R. F. E., Fox, C., Root, D. L., Hovy, E., & Lewis, R. (2015c). The C@merata task at MediaEval 2015: Natural language queries on classical music scores. In *Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/Paper12.pdf. Accessed 4 Mar 2017.

Sutcliffe, R. F. E., & Liem, C. (2017). Capturing the meaning of complex texts about music. In *Proceedings of ISMIR 2017, Suzhou, China.* https://ismir2017.smcnus.org/lbds/Sutcliffe2017.pdf. Accessed 1 Dec 2017.

Sutcliffe, R. F. E., Ó Maidín, D. S., & Hovy E. (2017). The C@merata task at MediaEval 2017: Natural language queries about music, their JSON representations, and matching passages in MusicXML scores. In *Proceedings of the MediaEval 2017 workshop, Trinity College Dublin, Ireland, September 13–15 2017*. http://ceur-ws.org/Vol-1984/Mediaeval_2017_paper_35.pdf. Accessed 2 Nov 2017.

Sutcliffe, R., Peñas, A., Hovy, E., Forner, P., Rodrigo, A., Forascu, C., Benajiba, Y., & Osenova, P. (2013). Overview of QA4MRE main task at CLEF 2013. online working notes. In *CLEF 2013 evaluation labs and workshop, 23–26 September 2013, Valencia, Spain*. http://csee.essex.ac.uk/staff/rsutcl/CLEF2013wn-QA4MRE-SutcliffeEt2013.pdf. Accessed 4 Mar 2017.

Tata, S., & Di Eugenio, B. (2010). Generating fine-grained reviews of songs from album reviews. In *Proceedings of the 48th ACL Annual Meeting, July* (pp. 1376–1385).

Tsaptsinos, A. (2017). Lyrics-based music genre classification using a hierarchical attention network. In *Proceedings of ISMIR, Suzhou, China, 2017.*

van Rijsbergen, K. J. (1979). *Information retrieval*. London, UK: Butterworth. http://www.dcs.gla.ac.uk/Keith/Preface.html. Accessed 24 Jan 2017.

Viglianti, R. (2015). *Enhancing music notation addressability*. http://mith.umd.edu/research/project/enhancing-music-notation-addressability/. Accessed 10 Mar 2017.

Voorhees, E. M. (2002). Overview of the TREC 2002 question answering track. In *Text retrieval conference (TREC)*. http://trec.nist.gov/pubs/trec11/papers/QA11.pdf. Accessed 15 Mar 2017.

Wan, S. (2014a). The CLAS system at the MediaEval 2014 C@merata task. In *Proceedings of the MediaEval 2014 workshop Barcelona, Catalunya, Spain, October 16–17, 2014.* http://ceur-ws.org/Vol-1263/mediaeval2014_submission_49.pdf. Accessed 12 Apr 2017.

Wan, S. (2014b). A description of the CLAS system at C@merata 2014. In *Proceedings of the C@merata task at MediaEval 2015.* http://csee.essex.ac.uk/camerata/clas_wan_long_14.pdf. Accessed 12 Apr 2017.

Wan, S. (2015a). CLAS at the MediaEval 2015 C@merata *task. In Proceedings of the MediaEval 2015 workshop, Wurzen, Germany, September 14–15 2015*. http://ceur-ws.org/Vol-1436/Paper85.pdf. Accessed 12 Apr 2017.

Wan, S. (2015b). CLAS at the C@merata 2015 task: Using unification between lexico-semantic query features and musical event metadata. In *Proceedings of the C@merata task at MediaEval 2014.* http://csee.essex.ac.uk/camerata/clas_wan_long_15.pdf. Accessed 12 Apr 2017.

Wang, Y., Kan, M., Nwe, T., Shenoy, A., & Yin, J. (2004). LyricAlly: Automatic synchronization of acoustic musical signals and textual lyrics. In *Proceedings of MM'04, New York, October*.

Whitman, B., & Ellis, D. (2004). Automatic record reviews. In *Proceedings of ISMIR.*

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *NAACL-HLT (*pp. 1480–1489).