CHARACTERISATION OF HARMONY WITH INDUCTIVE LOGIC PROGRAMMING

Amélie Anglade, Simon Dixon
Queen Mary University of London
Centre for Digital Music
{amelie.anglade, simon.dixon}@elec.qmul.ac.uk

ABSTRACT

We present an approach for the automatic characterisation of the harmony of song sets making use of relational induction of logical rules. We analyse manually annotated chord data available in RDF and interlinked with web identifiers for chords which themselves give access to the root, bass, component intervals of the chords. We pre-process these data to obtain high-level information such as chord category, degree and intervals between chords before passing them to an Inductive Logic Programming software which extracts the harmony rules underlying them. This framework is tested over the Beatles songs and the Real Book songs. It generates a total over several experiments of 12,450 harmony rules characterising and differentiating the Real Book (jazz) songs and the Beatles' (pop) music. Encouragingly, a preliminary analysis of the most common rules reveals a list of well-known pop and jazz patterns that could be completed by a more in depth analysis of the other rules.

1 INTRODUCTION

The explosion of the size of personal and commercial music collections has left both content providers and customers with a common difficulty: organising their huge musical libraries in such a way that each song can be easily retrieved, recommended and included in a playlist with similar songs. Most of the metadata provided are hand-annotated by experts such as in AllMusicGuide [1] or the result of a community effort such as in MusicBrainz [2]. Because classifying large amounts of data is expensive and/or time-consuming, people are gaining interest in the automatic characterisation of songs or groups of songs. Songs can be characterised by their rhythm, harmony, structure, instrumentation, etc. In this article we present the first step towards a framework able to automatically induce rules characterising songs by various musical phenomena. For this study we are interested in the automatic extraction of harmony patterns.

Many of the existing approaches for pattern extraction and recognition make use of statistical descriptions [14]. However some of the attempts to automatically build descriptors of music explore the idea of using logical rules [6]. Such rules could be expressed as follows:

$$(X_1 \quad X_2 ... X_n) \quad \textit{Musical Phenomenon} \qquad (1)$$

where the X_i are structured descriptions of local musical content and *Musical Phenomenon* is the high level property we are interested in. The logical framework offers several advantages over the statistical framework, for example temporal relations between local musical events can easily be expressed. Moreover, logical rules are human-readable. Thus, automatically extracted musical patterns expressed in logical formulae can be transmitted as they are to musicologists who can in turn analyse them.

To induce such logical rules we can either use statistical methods such as the C4.5 algorithm, a statistical decision tree learning algorithm, or relational methods such as Inductive Logic Programming (ILP) [5]. We choose to focus on relational induction of harmony rules with ILP. To test our framework we study the chord sequences of the Beatles and Real Book songs starting from a symbolic representation of these songs. However, the primary focus of this paper is on methodology and knowledge representation, rather than on the presentation of new musical knowledge extracted by the system.

The paper is organised as follows: In Section 2, we review some existing studies using logical rules for MIR. In Section 3 we explain our methodology to automatically extract logical harmony rules from manually annotated chords. In Section 4 the details and results of our automatic analysis of the Beatles and Real Book songs with ILP are presented and discussed before concluding in Section 5.

2 RELATED WORK

We now review some previous work in which automatically induced logical rules are used for musical retrieval tasks.

In [6] logical rules for characterising melody in MIDI files are built with random forest classifiers, a statistical approach. A pattern-based first order inductive system called PAL set up by Morales [4] learns counterpoint rules. The

[1] http://www.allmusic.com/
[2] http://musicbrainz.org/

system looks for patterns in the notes (described by their tone, height and voice) using a background knowledge restricted to the classification of intervals between two notes into perfect or imperfect consonant and dissonant, valid and invalid intervals.

Also many research projects have focused on automatically building expressive performance rules. For instance Widmer [16] develops an inductive rule learning system identifying both relevant and unknown rules of expressive performance from MIDI recordings of Mozart's sonatas performed by different pianists. His rules are based on the tempo, dynamics and articulation properties of the notes. Dovey [1] analyses and extracts rules from piano performances of Rachmaninoff using ILP. Subsequently, Van Baelen and De Readt [13] extend this work by transforming the characterisation rules into generative rules for MIDI.

Finally several musical rule derivation studies were conducted by Ramirez et al. [8, 10, 9]. In [8] an ILP based application learns rules in popular music harmonisation. They are constructed at a bar level (and not at a note level) to capture chord patterns. The structure (i.e. musical phrases) of the songs given as examples is manually annotated, which provides the system with a rich background knowledge containing not only local but also global information. Subsequently, Ramirez et al. [10] study Jazz performance from audio data with an ILP system. The system induces rules related to the duration transformation, onset deviation and energy of a note and the alteration of the score melody by adding or deleting notes. A background knowledge composed of information about the neighboring notes and the Narmour group(s) to which each note belongs is provided to the system. Finally, Ramirez et al. [9] implements a framework which analyses classical violin performance by means of both an ILP technique (the relational decision tree learner called TILDE) and a numerical method. Another component then uses these results to synthesise expressive performance from unexpressive melody descriptions.

3 METHODOLOGY

In search of chord idioms, Mauch et al. [3] made an inventory of chord sequences present in the Real Book and in the Beatles' studio albums. Their approach is entirely statistical and resulted in an exhaustive list of chord sequences together with their relative frequencies. To compare the results of our relational methodology with their results obtained with a statistical method we examine RDF (Resource Description Framework) descriptions of the two manually annotated collections from [3]:

- Harte's transcriptions [2] of the 180 songs featured on the Beatles' studio albums,
- transcriptions of 244 Jazz standards from the Real Book [15] [3].

These transcriptions constitute a compact symbolic representation of the songs: the chords are manually labelled in a jazz/pop/rock shorthand fashion and their start and end times are provided.

The steps to extract harmony rules from these songs transcriptions are summarised as follows: First the RDF representation of the harmonic events is pre-processed and transcribed into a logic programming format that can be understood by an Inductive Logic Programming system. This logic programming representation is passed to the ILP software Aleph [11] which induces the logical harmony rules underlying the harmonic events.

3.1 Harmonic content description

The RDF files describing the Beatles and Real songs we study contain a structured representation of the harmonic events based on the Music Ontology [7]. Each harmonic event (or chord) is associated with a start time, an end time and a web identifier we can crawl to get the RDF description of the chord based on the Chord Ontology [4].

We implemented a RDF chord parser to transcribe RDF chord representation into Prolog files that can be directly given as input to Aleph. For each of these chords it extracts the root note, bass note, component intervals, start time and end time from the RDF description. It then computes the chord category and degree of a chord and the root interval and bass interval between two consecutive chords. Intervals are measured upwards.

For this study we limit the chord categories (or chord types) to 'Major', 'minor', 'augmented', 'diminished', 'suspended', 'Dominant', 'neutral' (when the 3rd is neither present nor suspended) and 'unknown' (for every chord that does not belong to the previous categories). For each chord, the intervals are analysed by the RDF chord parser which then assigns the chord to one of these categories. First it reduces the chord to a 7th chord and checks if this reduced chord is a dominant 7th, in which case the chord is labeled 'Dominant'. Otherwise the chord is reduced to a triad and the type of this triad is kept as the chord category.

The degrees are computed by our RDF chord parser using the current key. Key information was added by hand when available. We only had access to tonality information for the Beatles, so no degree details were added for the Real Book songs. For the Beatles we performed two studies: one without degree over the whole set of songs and one with degree in which only the songs where there was no key modulation were kept. We also filtered out the songs which were not tonal songs (i.e. songs that were not following major or minor scales). The remaining songs constitute 73.9% of the Beatles' songs.

[3] these RDF files are available on http://chordtranscriptions.net/

[4] http://purl.org/ontology/chord

Our sole interest is in sequences of chords between which there is a harmonic modification (i.e. at least the root, bass or chord category differs from one chord to the next one). Although harmonic rhythm is important we leave it for future work.

3.2 Rule induction with ILP

The inductive concept learning task of Inductive Logic Programming can be described as follows. We provide a set of examples E containing positive (set E^+) and negative (set E^-) examples of a concept C and a background knowledge B all expressed in logic programs. Given this we find a hypothesis H (also expressed in logic program form) such that every positive example e E^+ is covered by H B (completeness) and no negative example e E^- is covered by H B (consistency).

We are interested in chord sequences of length 4 as in Mauch et al. [3]. Four chord sequences is a typical phrase length for the studied corpora. This choice is also the result of an empirical process: we also studied shorter sequences, but the results were less interesting, i.e. characteristic of the corpus and presenting well-known patterns. For longer sequences, the extracted patterns are less general, i.e. have a smaller coverage and thus less characteristic of the corpus. The concept we want to characterise is the harmony of a set of songs e.g. all the Beatles songs, all the Real Book songs. Therefore the positive examples given to the ILP system are all the chord sequences of length 4 (predicate chord_prog_4/4) found in such a set of songs. These chord sequences overlap. For instance from a chord sequence of length 8 we extract 5 overlapping chord sequences of length 4. The background knowledge is composed of the descriptions of all the chords previously derived by the RDF chord parser.

In the ILP system we use to induce harmony rules (Aleph [11]), we can either provide negative examples of a concept (in our case, chord progressions of length 4 from another set of songs) or force Aleph to explain the positive examples using a well-designed negative example (we will refer to this mode as the *one negative example mode*). In the latter case our negative example consists of the first chord sequence of our corpus in which we exchanged the position of the first and second chords. It is a valid negative example because in our background knowledge each chord in each song is uniquely identified by a Prolog atom and the position of each individual chord relative to the other chords is stored. We found out that by limiting the set of negative examples to this very simple one we obtained a more complete set of rules than when using the *positive examples only mode* of Aleph which randomly generates a limited number of negative examples.

To generate hypotheses Aleph uses inverse entailment. It consists of selecting an uncovered example, saturating it to obtain a bottom clause and searching the space of clauses that subsumes this bottom clause in a top-down manner starting from the shortest clauses. Saturating an example means looking for all the facts that are true about this example (using the example itself and the background knowledge). The bottom clause is the disjunction of all these facts. The clause that covers the maximum number of examples and the minimum number of negative examples (i.e. which maximises a score function based on the number of positive and negative examples covered by this clause) is kept as a hypothesis. The examples covered by the found hypothesis are removed and the next uncovered example is selected to be saturated, and so on until no uncovered example is left. Finally Aleph returns a set of hypotheses that covers all the positive examples. This process is not deterministic. The set of generated rules depends on the order in which the examples are selected by Aleph (which is the order in which the examples are given to Aleph). So the resulting set of rules is only one of the sets of rules that could be induced from the set of examples. However since Aleph looks for the most general rules at each step, the final set of rules is a sufficient description of the data (it explains all chord sequences) and is non-redundant (no subset of the rules explains all the chord sequences). This minimal sufficient description of a data set could be very useful for classification purposes since only a few characteristics needs to be computed to classify a new example. This is one of the advantages of our method against the purely statistical method employed in [3] which only computes the frequencies of each chord sequence and does not try to build a sufficient model of the corpora.

To obtain meaningful rules we also constrain Aleph to look for a hypothesis explaining the chord progressions in terms of root note progressions (root_prog_4/8), bass note progressions (bassNote_prog_4/8), chord category progressions (category_prog_4/8), root interval progressions (rootInterval_prog_3/7), bass interval progressions (bassInterval_prog_3/7) and degree progressions (degree_prog_4/8).

4 EXPERIMENTS AND RESULTS

4.1 Independent characterisation of the Beatles and Real Book chord sequences

We run two experiments. In the first experiment we want to characterise the chord sequences present in the Beatles' songs and compare them to the chord sequences present in the Real Book songs. Therefore we extract all the chord sequences of length 4 in the Beatles' tonal songs with no modulation (10,096 chord sequences), all the chord sequences of length 4 in all the Beatles' songs (13,593 chord sequences) and all the chord sequences of length 4 from the Real Book songs (23,677 chord sequences). Then for each of these sets

Rule	C_1	C_2
1. maj maj maj maj	4752 (35%)	3951 (39%)
2. maj maj maj min	632 (4.65%)	431 (4.27%)
3. min maj maj maj	628 (4.62%)	448 (4.44%)
4. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf5th}} \bullet \xrightarrow{\text{perf4th}} \bullet$	586 (4.31%)	-
5. $\bullet \xrightarrow{\text{/perfU}} \bullet \xrightarrow{\text{/perfU}} \bullet \xrightarrow{\text{/perfU}} \bullet$	584 (4.30%)	-
6. maj min maj maj	522 (3.84%)	384 (3.80%)
7. maj maj min maj	494 (3.63%)	363 (3.60%)
8. $\bullet \xrightarrow{\text{/perf5th}} \bullet \xrightarrow{\text{/perf4th}} \bullet \xrightarrow{\text{/perf5th}} \bullet$	463 (3.41%)	346 (3.43%)
9. maj maj min min	344 (2.53%)	217 (2.15%)
10. $\bullet \xrightarrow{\text{perfU}} \bullet \xrightarrow{\text{perfU}} \bullet \xrightarrow{\text{perfU}} \bullet$	336 (2.47%)	237 (2.38%)
11. min min maj maj	331 (2.44%)	216 (2.14%)
12. maj min min maj	308 (2.27%)	197 (1.95%)
13. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{maj2nd}} \bullet \xrightarrow{\text{perf4th}} \bullet$	260 (1.91%)	209 (2.07%)
14. $\bullet \xrightarrow{\text{maj2nd}} \bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf4th}} \bullet$	251 (1.85%)	195 (1.93%)

Table 1. Beatles harmony rules whose coverage is larger than 1.75%. C_1 and C_2 represent the positive coverage over all the Beatles songs and over the Beatles tonal songs with no modulation respectively. "perfU" means perfect unison.

of chord sequences we induce rules characterising them using the *one negative example mode* in Aleph.

Our system induces a set of 250 rules for each of the Beatles collections (tonal songs, all songs) and a set of 596 rules for the Real Book. The positive coverage of a rule is the number of positive examples covered by this rule. We want to consider only the patterns occurring in multiple songs (i.e. the ones characteristic of the corpus). For that we leave out the rules with a too small coverage (smaller than 0.5%). Because of space limitation we also only show the top rules for each experiment but a complete list of rules is available upon request. The top rules for our first experiment are shown in Tables 1 and 2.

For readability purposes we only show a compact representation of the body of rules:

- degrees are represented with roman numerals,
- "/ " refers to the bass note as in jazz chord notation,
- the intervals between roots (written first) or bass notes of the chords (symbolised by "/") are put on top of the arrows,
- a bullet symbolises the absence of information about some characteristics of the chord.

In accordance with conclusions in [3], some patterns extracted in these experiments are very common pop and jazz harmonic patterns. For instance, the Beatles rule with the highest coverage (more than a third of the chord sequences) is maj maj maj maj. The minor chord is the second

Rule	C
1. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf4th}} \bullet$	1861 (7.86%)
2. min dom min dom	969 (4.09%)
3. min dom maj min	727 (3.07%)
4. dom min dom min	726 (3.07%)
5. min min min min	708 (2.99%)
6. dom dom dom dom	674 (2.85%)
7. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perfU}} \bullet$	615 (2.60%)
8. $\bullet \xrightarrow{\text{maj6th}} \bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf4th}} \bullet$	611 (2.58%)
9. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf5th}} \bullet \xrightarrow{\text{perf4th}} \bullet$	608 (2.57%)
10. dom min dom maj	594 (2.51%)
11. dom maj min dom	586 (2.47%)
12. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perfU}} \bullet \xrightarrow{\text{perf4th}} \bullet$	579 (2.45%)
13. $\bullet \xrightarrow{\text{/maj6th}} \bullet \xrightarrow{\text{/perf4th}} \bullet \xrightarrow{\text{/perf4th}} \bullet$	547 (2.31%)
14. maj min dom maj	478 (2.02%)
15. $\bullet \xrightarrow{\text{/maj7th}} \bullet \xrightarrow{\text{/perf4th}} \bullet \xrightarrow{\text{/perf4th}} \bullet$	477 (2.01%)
16. $\bullet \xrightarrow{\text{/perf4th}} \bullet \xrightarrow{\text{/maj6th}} \bullet \xrightarrow{\text{/perf4th}} \bullet$	440 (1.86%)
17. $\bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{perf4th}} \bullet \xrightarrow{\text{maj6th}} \bullet$	436 (1.84%)
18. min dom maj dom	424 (1.79%)

Table 2. Real Book harmony rules whose coverage is larger than 1.75%. C is the positive coverage.

	Rule	C_1	C_2
1.	maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{perf5th}}$ maj $\xrightarrow{\text{perf4th}}$ maj	3.13%	3.79%
	I maj IV maj I maj IV maj	-	2.47%
	V maj I maj V maj I maj	-	1.00%
2.	maj $\xrightarrow{\text{perf5th}}$ maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{perf5th}}$ maj	2.94%	3.61%
	IV maj I maj IV maj I maj	-	2.43%
	I maj V maj I maj V maj	-	0.84%
3.	maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{maj2nd}}$ maj $\xrightarrow{\text{perf4th}}$ maj	1.38%	1.75%
	I maj IV maj V maj I maj	-	1.59%
4.	maj $\xrightarrow{\text{maj2nd}}$ maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{perf4th}}$ maj	1.21%	1.47%
	IV maj V maj I maj IV maj	-	1.15%
5.	maj $\xrightarrow{\text{perf5th}}$ maj $\xrightarrow{\text{min7th}}$ maj $\xrightarrow{\text{perf5th}}$ maj	1.04%	1.28%
	I maj V maj IV maj I maj	-	0.69%
	IV maj $\rightarrow$ I maj $\rightarrow$ bVII maj $\rightarrow$ IV maj	-	0.52%
6.	maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{maj2nd}}$ maj	0.93%	1.11%
	V maj I maj IV maj V maj	-	1.03%
7.	maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{perf4th}}$ maj $\xrightarrow{\text{perf5th}}$ maj	0.91%	1.09%
	V maj I maj IV maj I maj	-	0.83%

Table 3. Beatles root interval and chord category rules (whose coverage is larger than 1%) and the associated degree and chord category rules. C_1 and C_2 represent the positive coverage over all the Beatles songs and over the Beatles tonal songs with no modulation respectively.

Rule	C
1. maj $\overset{\text{maj6th}}{-}$ min $\overset{\text{perf4th}}{-}$ min $\overset{\text{perf4th}}{-}$ dom	190 (0.80%)
2. dom $\overset{\text{perf4th}}{-}$ min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf4th}}{-}$ maj	176 (0.74%)
3. min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf4th}}{-}$ min $\overset{\text{perf4th}}{-}$ dom	174 (0.73%)
4. min $\overset{\text{perf4th}}{-}$ min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf4th}}{-}$ maj	171 (0.72%)
5. min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf4th}}{-}$ maj $\overset{\text{maj6th}}{-}$ min	170 (0.72%)
6. dom $\overset{\text{perfU}}{-}$ min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf4th}}{-}$ maj	133 (0.56%)
7. maj $\overset{\text{maj2nd}}{-}$ min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf4th}}{-}$ maj	126 (0.53%)
8. min $\overset{\text{perf4th}}{-}$ dom $\overset{\text{perf5th}}{-}$ min $\overset{\text{perf4th}}{-}$ dom	124 (0.52%)
9. min $\overset{\text{perfU}}{-}$ min $\overset{\text{perfU}}{-}$ min $\overset{\text{perfU}}{-}$ min	124 (0.52%)
10. dom $\overset{\text{perf4th}}{-}$ maj $\overset{\text{maj6th}}{-}$ min $\overset{\text{perf4th}}{-}$ min	121 (0.51%)

Table 4. Top ten Real Book harmony rules when considering root interval progressions and chord category progressions. C is the positive coverage.

most frequent chord category in the Beatles and the dominant chord ranks quite low in the chord category rules (rule 25 not shown here). For the Real Book, the rule with the highest coverage is $\bullet \overset{\text{perf4th}}{-} \bullet \overset{\text{perf4th}}{-} \bullet \overset{\text{perf4th}}{-} \bullet$. An interpretation of this rule is that for most of the instances it certainly is ii-V-I-IV (a very common jazz chord progression). Another common jazz chord progression, I-VI-II-V (often used as “turnaround” in jazz), is captured by rule 8 in Table 2. Moreover the dominant chord is the most frequent chord category in the Real Book which clearly distinguishes the jazz standards of the Real Book from the pop songs of the Beatles.

Note that due to the fact that the chord sequences overlap and due to the cyclic nature of some of the pop and jazz songs, many rules are not independent. For instance rules 2, 3, 6 and 7 in Table 1 can represent the same chord sequence maj-maj-maj-min repeated several times.

Moreover Aleph can also derive rules that contain some degree information. For that we constrain Aleph to derive rules about the intervals between the chord roots which we believe capture degree patterns. The top root interval and category rules for each corpus are presented in Tables 3 and 4. Furthermore, since we have key information for some of the Beatles songs we can actually obtain degree rules for them and an analysis of the degree rules allows us to match each root interval rule (with no tonal centre information) with the degree rules which are covered by it. The result of this matching process between degree and root interval rules is presented in Table 3 (top rules only). So for instance in Table 3 the instances of the root interval rule 5:

$$\text{maj} \overset{\text{perf5th}}{-} \text{maj} \overset{\text{min7th}}{-} \text{maj} \overset{\text{perf5th}}{-} \text{maj}$$

Rule	C
1. maj $\overset{\text{perf4th}}{-}$ maj $\overset{\text{maj2nd}}{-}$ maj $\overset{\text{perf4th}}{-}$ maj	188 (1.38%)
2. maj $\overset{\text{maj2nd}}{-}$ maj $\overset{\text{perf4th}}{-}$ maj $\overset{\text{perf4th}}{-}$ maj	165 (1.21%)
3. maj $\overset{\text{perf5th}}{-}$ maj $\overset{\text{min7th}}{-}$ maj $\overset{\text{perf5th}}{-}$ maj	141 (1.04%)
4. maj $\overset{\text{perf4th}}{-}$ maj $\overset{\text{perf4th}}{-}$ maj $\overset{\text{maj2nd}}{-}$ maj	126 (0.93%)
5. A maj D maj A maj D maj	114 (0.84%)
6. maj$/_A$ maj$/_A$ maj$/_A$ maj$/_A$	110 (0.81%)
7. maj $\overset{\text{min7th}}{-}$ maj $\overset{\text{perf5th}}{-}$ maj $\overset{\text{perf5th}}{-}$ maj	108 (0.79%)
8. maj $\overset{\text{perf5th}}{-}$ maj $\overset{\text{perf5th}}{-}$ maj $\overset{\text{min7th}}{-}$ maj	102 (0.75%)
9. maj $\overset{\text{perfU}}{-}$ maj $\overset{\text{perf4th}}{-}$ maj $\overset{\text{perf5th}}{-}$ maj	99 (0.73%)
10. D maj G maj D maj G maj	92 (0.68%)

Table 5. Top ten Beatles harmony rules when the Real Book is taken as negative example. C is the positive coverage.

are for 54% of them instances of the degree rule:

I maj V maj IV maj I maj

and for 41%, instances of the degree rule:

IV maj I maj bVII maj IV maj

4.2 Characterisation of the Beatles vs. Real Book songs

For the second experiment we want to know the Beatles chord sequences that are not present in the Real Book. Aleph is provided with all the Beatles chord sequences of length 4 as positive examples and all the Real Book chord sequences of length 4 as negative examples. It returns 1679 rules which characterise all the chord sequences that only appear in the Beatles songs. The top ten rules are shown in Table 5. Some of these rules are correlated. For instance the 3 chord cyclic pattern I-IV-V-I-IV-V-I..., very common in the early compositions of the Beatles (see for instance the song *Please Please Me* of the album *Please Please Me*), is covered by rules 1, 2 and 4. Similarly the cyclic pattern I-V-IV-I-V-IV-I... is covered by rules 3, 7 and 8. Note also that the “back and forth” pattern between the first and fourth degree mentioned in [3] and identified in rule 1 of Table 3 appears in rules 5 and 10 of Table 5.

As in the previous experiment we also try to characterise the chord sequences in terms of root intervals and chord categories and obtain a set of 1520 rules which are not presented here.

4.3 Considerations about the size of the corpora and the computation time

Such an ILP approach has never been applied on such a scale: we dealt with data sets a musicologist would typically be interested in studying (unified corpora of songs commonly accepted as representative of a composer/band/genre).

Although ILP systems are known to usually be resource intensive, the computation time of the ILP system was not a limiting factor in this work. Aleph computed all the rules in less than a minute on a regular desktop computer. We see our framework as a useful tool for musicologists since manual harmonic annotation and analysis of the whole Beatles corpus takes several years of musicological work whereas the automatic extraction of the chord progression patterns using ILP takes only seconds, allowing the user to concentrate on the interpretation of the results.

5 CONCLUSION AND FUTURE WORK

In this paper we presented a framework to automatically extract harmony rules from manually labelled chords using Inductive Logic Programing. We also gave an overview of the most common harmony rules extracted on the Beatles and Real Book songs using this framework. This first analysis shows that very common jazz and pop patterns are present in these rules. We hope that an in-depth musicological analysis of the other rules will reveal other less common and more specific harmonic patterns. We identified patterns listed in [3] but our methodology does more than listing and counting chord sequences, it also builds a minimal rule set which describes a data set. We believe this technique can be used by musicologists to automatically characterise the harmony of large sets of songs in few seconds.

Furthermore, since our system builds a sufficient model of a data set in future work we intend to test whether such logical rules can efficiently be used for classification and clustering purposes. We also plan to use this technique to characterise other musical phenomena such as rhythm, melody, structure from symbolic data. In order to deal with more data and to avoid the time consuming task of manual annotation of collections we intend to use automatic symbolic analysis systems such as Melisma [12]. A further step will be to adapt our ILP framework to audio data.

6 ACKNOWLEDGMENTS

This work is part of the OMRAS2 project funded by EPSRC grant EP/E017614/1.

7 REFERENCES

[1] M. J. Dovey. Analysis of Rachmaninoff's piano performances using inductive logic programming. In *Proceedings of the European Conference on Machine Learning*, pages 279–282. Springer-Verlag, 1995.

[2] C. Harte, M. Sandler, S. Abdallah, and E. Gómez. Symbolic representation of musical chords: a proposed syntax for text annotations. In *Proceedings of ISMIR 2005*, pages 66–71, London, UK, 2005.

[3] M. Mauch, S. Dixon, C. Harte, M. Casey, and B. Fields. Discovering chord idioms through Beatles and Real Book songs. In *Proceedings of ISMIR 2007*, pages 255–258, Vienna, Austria, 2007.

[4] E. F. Morales. PAL: A pattern-based first-order inductive system. *Machine Learning*, 26:227–252, 1997.

[5] S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[6] P. J. Ponce de León, D. Rizo, and J. M. Iñesta. Towards a human-friendly melody characterization by automatically induced rules. In *Proceedings of ISMIR 2007*, pages 437–440, Vienna, Austria, 2007.

[7] Y. Raimond, S. Abdallah, and M. Sandler. The Music Ontology. In *Proceedings of ISMIR 2007*, pages 417–422, Vienna, Austria, 2007.

[8] R. Ramirez. Inducing musical rules with ILP. In *Proceedings of the International Conference on Logic Programming*, pages 502–504. Springer-Verlag (LNCS)), 2003.

[9] R. Ramirez and A. Hazan. A tool for generating and explaining expressive music performances of monophonic jazz melodies. *International Journal on Artificial Intelligence Tools*, 15(4):673–691, 2006.

[10] R. Ramirez, A. Hazan, E. Gómez, and E. Maestre. Understanding expressive transformations in saxophone jazz performances using inductive machine learning. In *Proceedings Sound and Music Computing International Conference*, Paris, 2004.

[11] A. Srinivasan. The Aleph Manual (version 4), 2003.

[12] D. Temperley and D. Sleator. Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27, 1999.

[13] E. Van Baelen and L. De Raedt. Analysis and prediction of piano performances using Inductive Logic Programming. In *Proceedings of the International Conference in Inductive Logic Programming*, pages 55–71, 1996.

[14] P. van Kranenburg. Composer attribution by quantifying compositional strategies. In *Proceedings of ISMIR 2006*, Victoria, Canada, 2006.

[15] various. *The Real Book*. Hal Leonard Corporation, 6th edition, 2004.

[16] G. Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003.