HARMTRACE: IMPROVING HARMONIC SIMILARITY ESTIMATION USING FUNCTIONAL HARMONY ANALYSIS

W. Bas de Haas
Utrecht University
W.B.deHaas@uu.nl

José Pedro Magalhães
Utrecht University
J.P.Magalhaes@uu.nl

Remco C. Veltkamp
Utrecht University
R.C.Veltkamp@uu.nl

Frans Wiering
Utrecht University
F.Wiering@uu.nl

ABSTRACT

Harmony theory has been essential in composing, analysing, and performing music for centuries. Since Western tonal harmony exhibits a considerable amount of structure and regularity, it lends itself to formalisation. In this paper we present HARMTRACE, a system that, given a sequence of symbolic chord labels, automatically derives the harmonic function of a chord in its tonal context. Among other applications, these functional annotations can be used to improve the estimation of harmonic similarity in a local alignment of two annotated chord sequences. We evaluate HARMTRACE and three other harmonic similarity measures on a corpus of 5,028 chord sequences that contains harmonically related pieces. The results show that HARMTRACE outperforms all three other similarity measures, and that information about the harmonic function of a chord improves the estimation of harmonic similarity between two chord sequences.

1. INTRODUCTION

With the rapid expansion of digital repositories of music, such as iTunes, Spotify, last.fm, and the like, efficient methods to provide content-based access to this kind of music repositories have become increasingly important. To be able to cluster documents, a notion of the similarity between these documents is essential. Hence, within Music Information Retrieval (MIR), the development of musical similarity measures plays a prominent role. Music can be related in many different aspects, e.g. melody, genre, rhythm, etc.; this paper focuses on similarity of musical harmony. Music retrieval based on harmony offers obvious benefits: it allows for finding cover songs (especially when melodies vary), songs of a certain family (like Blues or Rhythm Changes), or variations over a theme in baroque music, to name a few.

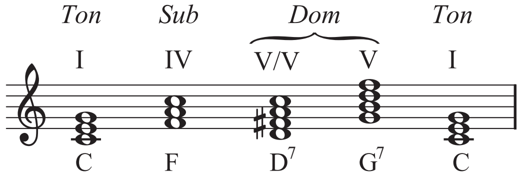

Figure 1. A typical chord sequence. The chord labels are printed below the score, and the scale degrees and functional analysis above the score. Tonic, dominant, and subsominant are denoted with *Ton*, *Dom*, and *sub*, respectively.

To be able to understand why two chord sequences are harmonically related, we believe it is important to examine chords not only in isolation but also the context in which they occur. For this, we draw greatly on classical and jazz harmony theory. In the last decades, many music theorists have studied tonal harmony and observed that within a sequence not every chord is equally important. This suggests that tonal harmony is organised hierarchically. Within a sequence of chords, some chords can be removed leaving the global harmony structure intact, while removing other chords can significantly change how the chord sequence is perceived. For example in Figure 1, the D^7 chord could be removed without changing the general structure of the harmony, while removing the G^7 or the C at the end would change the harmony structure. This suggests that chords can have different functional roles, and therefore different importance.

Nowadays there is a rich body of literature that aims to explain the order and regularities in Western tonal harmony, and various ways to analyse the *function* of a chord in its tonal context have been proposed [9, 14, 18]. Unfortunately, the majority of these theories are formulated rather informally and lack descriptions with mathematical precision or computational executability. Although there are exceptions, like the Tonal Pitch Space model [8] and David Temperley's Melisma [22], the lack of mathematical precision has hampered the successful application of harmony models to practical MIR related tasks, such as automatic analysis, similarity estimation, content-based retrieval, or the improvement of low-level feature extraction.

Contribution We present HarmTrace [1] , a system for analysing Western tonal harmony and determining harmonic similarity, implemented robustly and efficiently in the pure, type-safe functional programming language Haskell. It is flexible, in the sense that it can be easily adapted and maintained, robust against noisy data, and capable of displaying harmonic analyses in a clear way. We evaluate the retrieval performance of HarmTrace by comparing it to a baseline alignment system and to two earlier approaches to harmonic similarity in a retrieval experiment, using a corpus of 5,028 chord sequences. The results show that HarmTrace outperforms all other harmonic similarity measures and that exploiting knowledge about the harmonic function of a chord improves retrieval performance.

The rest of this paper is organised as follows. After a review of related work in Section 2, we explain how an automatic harmony analysis is performed by a music theoretically founded knowledge system of tonal harmony (Section 3). Next, we define harmonic similarity of two sequences of annotated chords as the maximum local alignment score (Section 4). In Section 5 we compare the retrieval performance of HarmTrace to three other harmonic similarity measures. Finally, we conclude the paper with a short discussion on harmonic similarity and pointing out directions for future research (Section 6).

2. RELATED WORK

Grammatical models of tonal harmony harmony have a long history in music research, e.g. [9, 15, 20]. The harmony model of HarmTrace is based on the generative formalism proposed by Rohrmeier [16, 17]. He models tonal harmony as an elaborate recursive context-free grammar (CFG). His model extends ideas of the *Generative Theory of Tonal Music* (GTTM) [9] and Schenkerian Analysis [18], and captures form, theoretical harmonic function [14], phrasing, and modulation. De Haas et al. [4] performed a first attempt at implementing Rohrmeier's grammar and using it for defining harmonic similarity. HarmTrace transports these ideas to a functional setting, solving many of the typical problems accociated with context free parsing.

There exist other systems that address polyphonic music similarity, but generally these are embedded into larger retrieval systems and take audio or score information as input, e.g. [13]. We are aware of two other systems that focus solely on harmonic similarity and compute similarity values from textual chord descriptions: the Tonal Pitch Step Distance (TPSD) [5], and the Chord Sequence Alignment System
(CSAS) [6]. A benefit of evaluating only a similarity measure is that errors caused by the feature extraction or chord labelling methods do not influence the retrieval evaluation. The TPSD and CSAS are compared elaborately in [3]; we introduce them briefly here.

The TPSD uses Lerdahl's [8] Tonal Pitch Space (TPS) as its main musical model. TPS is a model of tonality that fits musicological intuitions, correlates well with empirical findings from music cognition, and can be used to calculate a distance between two arbitrary chords. The TPS model takes into account the number of steps on the circle of fifths between the roots of the chords, and the amount of overlap between the chord structures of the two chords and their relation to the global key.

The general idea behind the TPSD is to use the TPS to compare the change of perceived chordal distance to the tonic over time. For every chord, the TPS distance to the key of the sequence is calculated, resulting in a step function. Next, the distance between two chord sequences is defined as the minimal area between the two step functions over all possible horizontal circular shifts. To prevent that longer sequences yield larger distances, the score is normalized by the duration of the shortest song.

The CSAS [6] is based on local alignment: by performing elementary deletion, insertion, and substitution operations, one chord sequence is transformed into the other. The actual similarity value is defined as the total sum of all edit operations at all beat positions. To improve the retrieval performance of the classical alignment approach, Hanna et al. experimented with various musical data representations and substitution functions. They found a key-relative representation, based on the interval between the root of the chord and the key, to work well and preferred substituting only when the chord root and triad were not identical. In the experiments in [3] the CSAS outperformed the TPSD in 4 of the 6 tasks.

3. HARMONY MODEL

The HarmTrace harmony model implements and extends the ideas of Rohrmeier [16, 17]. However, HarmTrace differs from Rohrmeier's grammar in several aspects. Rohrmeier's model is more elaborate, as it includes phrasing and modulation. However, we believe that modulation and phrasing cannot be implemented as context-free rules in the way Rohrmeier formulates them. Rohrmeier's CFG allows for modulating into any key at any point in a sequence; from an implementation perspective, this would generate too many ambiguous solutions for a single sequence of chords. Furthermore, whereas Rohrmeier's grammar aims to explain the core rules of tonal harmony, our model exhibits a bias towards jazz harmony, due to the nature of the data used in Section 5.

We model tonal harmony as a complex Haskell datatype. To explain our model in a clear manner, that does not re-

[1] Harmony Analysis and Retrieval of Music with Type-level Representations of Abstract Chords Entities

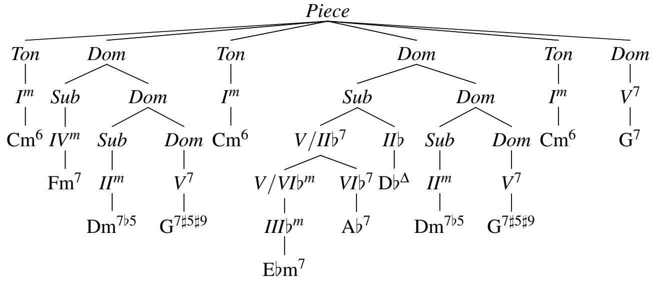
Piece
Ton Dom Ton Dom Ton Dom
I^m Sub Dom I^m Sub Dom I^m V^7
Cm^6 IV^m Sub Dom Cm^6 V/II♭^7 II♭ Sub Dom Cm^6 G^7
Fm^7 II^m V^7 V/VI♭^m VI♭^7 D♭^Δ II^m V^7
Dm^7♭5 G^7♯5♯9 III♭^m A♭^7 Dm^7♭5 G^7♯5♯9
E♭m^7

Figure 2. An analysis of the jazz standard *Blue Bossa* in C minor. Every chord belongs to a Tonic, Dominant, or Subdominant category (*Ton*, *Dom*, or *Sub*) and the V/X^7 denote chains of secondary dominants.

quire elaborate knowledge of the Haskell programming language, we chose a syntax that closely resembles a (very constrained) CFG. A CFG defines a language: it accepts only combinations of words that are valid sequences of the language. A collection of Haskell datatypes can be viewed as a very powerful CFG: the type-checker accepts a combination of values if their structure matches the structure prescribed by the datatype, and rejects this combination if it does not. Within HARMTRACE, the chords are the values and the datatypes represent the relations between the structural elements in tonal harmony.

3.1 A model of Western tonal harmony

Figure 2 shows an example analysis as produced by HARMTRACE. We start by introducing a variable (denoted with bold font) for the mode of the key of the piece, which can be major or minor. The mode variable is used to parametrise all the specifications of our harmony model; some specifications hold for both modes (**m**), while other specifications hold only for the major (Maj) or minor mode (Min). The mode is displayed as a subscript, which we leave out when it is clear from the context. Currently, HARMTRACE cannot yet derive the key of the piece automatically. Hence, to be able to use key-relative representations, external information about the key of the piece is essential.

1 $Piece_{\mathbf{m}} \rightarrow Func^{+}_{\mathbf{m}}$

2 $Func_{\mathbf{m}} \rightarrow Ton_{\mathbf{m}} \mid Dom_{\mathbf{m}} \qquad \mathbf{m} \in \{\text{Maj}, \text{Min}\}$

3 $Dom_{\mathbf{m}} \rightarrow Sub_{\mathbf{m}}\ Dom_{\mathbf{m}}$

Spec. 1–3 define that a valid chord sequence, $Piece_{\mathbf{m}}$, consists of at least one and possibly more functional categories. A functional category classifies chords as being part of a tonic ($Ton_{\mathbf{m}}$), dominant ($Dom_{\mathbf{m}}$), or subdominant ($Sub_{\mathbf{m}}$) structure, where a subdominant must always precede a dominant. The order of the dominants and tonics is not constrained by the model, and they are not grouped into larger phrases.

4 $Ton_{\text{Maj}} \rightarrow I_{\text{Maj}} \mid I_{\text{Maj}}\ IV_{\text{Maj}}\ I_{\text{Maj}}$

5 $Ton_{\text{Min}} \rightarrow I^{m}_{\text{Min}} \mid I^{m}_{\text{Min}}\ IV^{m}_{\text{Min}}\ I^{m}_{\text{Min}}$

6 $Dom_{\mathbf{m}} \rightarrow V^{7}_{\mathbf{m}} \mid V_{\mathbf{m}} \qquad \mathbf{c} \in \{\varnothing, m, 7, 0\}$

7 $Sub_{\text{Maj}} \rightarrow IV^{m}_{\text{Maj}} \mid II^{m}_{\text{Maj}} \mid \ldots$

8 $Sub_{\text{Min}} \rightarrow IV_{\text{Min}} \mid II^{m}_{\text{Min}} \mid \ldots$

Spec. 4–8 translate dominants, tonics, and sub-dominants into scale degrees (denoted with Roman numerals). A scale degree is a datatype that is parametrised by a mode, a chord class, and the interval between the chord root and the key. The chord class is used to constrain the application of certain specifications, e.g. Spec. 13 and 14, and can represent the class of major (no superscript), minor (m), dominant seventh (7), and diminished seventh chords (0). A tonic translates into a first degree in both major and minor mode, albeit with a minor triad in the latter case, or it allows for initiation of a plagal cadence. A dominant type is converted into the fifth or seventh scale degree with a dominant or diminished class, respectively. Similarly, a sub-dominant is converted into the fourth or second degree.

9 $I_{\text{Maj}} \rightarrow$ `"C:maj" | "C:maj6" | "C:maj7" | ...`

10 $I^{m}_{\text{Min}} \rightarrow$ `"C:min" | "C:min7" | "C:min9" | ...`

11 $V^{7}_{\mathbf{m}} \rightarrow$ `"G:7" | "G:7(b9,13)" | "G:(#11)" | ...`

12 $VII^{0}_{\mathbf{m}} \rightarrow$ `"B:dim(bb7)"`

Finally, scale degrees are translated into the actual surface chords that are used as input for the model. The chord notation used is that of Harte et al. [7]. The conversions are trivial and illustrated by a small number of specifications above. The model uses a key-relative representation, and in Spec. 9–12 we used chords in the key of C. Hence, a I_{Maj} translates to the set of C chords with a major triad, optionally augmented with additional chord notes that do not make the chord minor or dominant. Similarly, V^{7}_{Maj} translates to all G chords with a major triad and a minor seventh, etc.

13 $X^{\mathbf{c}}_{\mathbf{m}} \rightarrow V/X^{7}_{\mathbf{m}}\ X^{\mathbf{c}}_{\mathbf{m}} \qquad \mathbf{c} \in \{\varnothing, m, 7, 0\}$

14 $X^{7}_{\mathbf{m}} \rightarrow V/X^{m}_{\mathbf{m}}\ X^{7}_{\mathbf{m}} \qquad X \in \{I, II\flat, II, \ldots, VII\}$

Spec. 13 accounts for the classical preparation of a scale degree by its secondary dominant, stating that every scale degree, independently of its mode, chord class, and root interval, can be preceded by a chord of the dominant class, one fifth up. The function V/X which transposes an arbitrary scale degree X a fifth up. Similarly, every scale degree of the dominant class can be prepared with the minor chord one fifth above (Spec. 14). These two specifications together allow for the derivation of the typical and prominently present ii-V motions in jazz harmony.

15 $X^{7}_{\mathbf{m}} \rightarrow V\flat/X^{7}_{\mathbf{m}}$

16 $X^{7}_{\mathbf{m}} \rightarrow II\flat/X^{0}_{\mathbf{m}}$

17 $X^0_{\mathbf{m}} \rightarrow III\flat / X^0_{\mathbf{m}}$

The harmony model in HARMTRACE further allows various scale degree transformations. Every dominant chord can be transformed into its tritone substitution with Spec. 15. This specification uses another transposition function $V\flat / X$ which transposes a scale degree X a diminished fifth—a tritone—up. Likewise, diminished seventh chords are treated as regular dominant seventh chords without a root and with a ♭9 (rule 16). For instance, an A♭^{0}, consisting of A♭, B, D, and F, is viewed as a G$^{7\flat 9}$, which consists of G, B, D, F, and A♭^{0}. An exceptional characteristic of diminished seventh chords—consisting only of notes separated by minor third intervals—is that they are completely symmetrical. Hence, a diminished seventh chord has four enharmonic equivalent chords that can be reached by transposing the chord a minor third up with the transposition function $III\flat / X$ (Spec. 17). Because we want the application of the Spec. 13–17 to terminate, we limit the number of recursive applications of these rules. For the technical details about how this is done, we refer to [10].

We have presented a condensed view on the core specifications of the model, but due to space limitation we had to omit some specification for diatonic chains of fifths, borrowings from the parallel mode and the Neapolitan chord (see Figure 2). For the full specification of the model we refer to [2] and to the code bundle found online. [2]

3.2 Parsing

Having a formal specification as a datatype, the next step is to define a parser to transform textual chord labels into values of our datatype. Writing a parser that parses labels into our datatype would normally mean writing tedious code that closely resembles the datatype specification. However, in Haskell we can use *datatype-generic programming* [3] techniques to avoid writing most of the repetitive portions of the code. Moreover, not only the parser can be derived automatically, but also a pretty-printer for displaying the harmony analysis in tree form, and functions for comparing these analyses. This makes the development and fine-tuning of the model much easier, as only the datatype specifications have to be changed, and the code adapts itself automatically. For technical details of the implementation and the generic programming techniques we refer to [10].

Because music is an ever changing, culturally dependent, and extremely diverse art form, we cannot hope to model all valid harmonic relations in our datatype. Furthermore, songs may contain mistakes or mistyped chords, perhaps feature extraction noise, or malformed data of dubious harmonic validity. This is problematic if we reject chord sequences that do not fit the grammatical specification without returning any information about harmony analysis. However, these problems often occur at a specific position in the piece and most of the song still makes sense. In HARMTRACE we use a parsing library [21] that features error-correction: chords that do not fit the structure are automatically deleted or preceded by inserted chords, according to heuristics computed from the grammar structure. For most songs, parsing proceeds with none or very few corrections. Songs with a very high error ratio denote bad input or wrong key assignment, which results in meaningless scale degrees.

Music, and harmony in particular, is intrinsically ambiguous. Hence, certain chords can have multiple meanings within a tonal context. This is reflected in the model above. We control the number of possible analyses by constraining the application of most specifications. Examples hereof are the restriction of secondary dominants to scale degrees of the dominant class, and limiting the number of possible recursive applications of the secondary dominant rule.

4. SIMILARITY ESTIMATION

After having obtained an harmonic analysis from our model, a chord is categorised as being part of either a dominant, sub-dominant, or tonic structure (Spec. 4–8). Furthermore, we also annotate whether a chord is part of secondary dominant preparation (Spec. 13–14) and label whether it has been transformed (Spec. 15–17). We hypothesise that these annotations are helpful in determining harmonic similarity. Hence, we represent an annotated chord as a quintuple of the following form: $(X, \mathbf{c}, \mathit{func}, \mathit{prep}, \mathit{trans})$, where X represents a scale degree, $\mathbf{c}$ a chord class (as defined in Section 3), *func* the functional category, *prep* the functional preparation, e.g. being part of a secondary dominant (V / X), and *trans* a scale degree transformation, e.g. a tritone or diminished seventh substitution. For estimating the similarity between two sequences of these annotated chords we calculate the alignment score obtained in a classical alignment procedure [19].

The quality of an alignment heavily depends on the insertion, deletions, match, and mismatch parameters. We use a constant insertion and deletion penalty of -2 and we define the similarity between the annotated chords as a function, $\mathit{sim}\ (a_i, b_j) \rightarrow [-1, 6]$, that takes a pair of chords, a_i and b_j, and returns an integer denoting the (dis-) similarity. Here i and j denote the beat position of a_i and b_j in the compared chord sequences A and B.

$$\begin{array}{l}
\mathit{sim}\ (X_1, \mathbf{c}_1, \mathit{func}_1, \mathit{prep}_1, \mathit{trans}_1)\ (X_2, \mathbf{c}_2, \mathit{func}_2, \mathit{prep}_2, \mathit{trans}_2) = \\
\quad \mathbf{if}\ X_1 \equiv X_2 \wedge \mathbf{c}_1 \equiv \mathbf{c}_2\ \mathbf{then}\ 2 + m_{prep} + m_{trans}\ \mathbf{else}\ -1 \\
\qquad \mathbf{where}\ m_{prep} = \mathit{sim}_{prep}\ (\mathit{Prep}_1, \mathit{Prep}_2) \\
\qquad\qquad\quad m_{trans} = \mathbf{if}\ \mathit{Trans}_1 \equiv \mathit{Trans}_2\ \mathbf{then}\ 1\ \mathbf{else}\ 0
\end{array}$$

Within *sim*, the function $\mathit{sim}_{prep} \rightarrow [0, 3]$ compares two possible scale degree preparations, returning 3 is the preparation is identical, 2 if both preparations involve the same fifth

[2] http://hackage.haskell.org/package/HarmTrace-0.7

[3] Not to be confused with regular polymorphism, as in Java generics.

jump, 1 if they are both a preparation, and 0 in all other cases.

The final similarity score is obtained by calculating the optimal alignment between two annotated chord sequences and normalising the alignment score. Because the prefix of an optimal alignment is also an optimal alignment, an optimal solution can be found by exploiting the dynamic programming paradigm. To ensure that the alignment is maximal, we construct an array T which stores the cumulative alignment score so far. T is filled by calculating the recurrence below for every combination of annotated chords in the sequence A and B in a standard dynamic programming procedure.

$$T[i,j] = \max \begin{cases} M[i,j-1]-2, \\ M[i-1,j]-2, \\ M[i-1,j-1]+sim(a_i,b_j), \\ 0 \end{cases}$$

The actual alignment can be obtained by keeping track of the path trough T, starting at $T[n,m]$, where n and m are the sizes of A and B, respectively. We obtain our final similarity measure, $\textsc{sim}(A,B) \rightarrow [0,1]$, by normalising the sum of alignment scores, $T[n,m]$, by the sizes of A and B:

$$\textsc{sim}(A,B) = \frac{T[n,m]}{n} \cdot \frac{T[n,m]}{m}$$

5. EVALUATION

To evaluate the effect of the HarmTrace harmony model on retrieval performance, we compare it to a baseline alignment system, named TriadAlign. In TriadAlign we use the exact same alignment code, but the similarity function for individual chords, *sim*, is replaced by sim^{triad} that does not use any additional model information.

$$\begin{aligned} & sim^{triad}\ (X_1, triad_1)\ (X_2, triad_2) = \\ & \qquad \textbf{if}\ X_1 \equiv X_2 \wedge triad_1 \equiv triad_2\ \textbf{then}\ 4\ \textbf{else}\ -1 \end{aligned}$$

Here, *triad* denotes only whether the chord is major or minor, and the X represents the scale degree, as defined in the previous sections. Note that the TriadAlign system is very similar to the CSAS, but uses slightly different parameters and normalises the alignment score.

We compare the retrieval performance of HarmTrace, TriadAlign, TPSD, and CSAS methods (see Section 2) in a retrieval experiment for which we use the same chord sequence corpus as in [3]. This corpus consists of 5,028 unique user-generated Band-in-a-Box files that are collected from the Internet. Band-in-a-Box [1] is a commercial software package for generating musical accompaniment based on a lead sheet. For extracting the chord label information from the Band-in-a-Box files we have extended software in [12].

	TPSD	CSAS	TriadAlign	HarmTrace
MAP	0.580	0.696	0.711	0.722

Table 1. The mean average precision of the rankings based on the compared similarity measures.

Within the corpus, 1,775 songs contain two or more similar versions, forming 691 classes of songs. Within a song class, songs have the same title and share a similar melody, but may differ in a number of ways. They may, for instance, differ in key and form, in the number of repetitions, or may simply use different chords at certain positions. Having multiple chord sequences describing the same song allows for setting up a *cover-song*-finding experiment. The title of the song is used as ground-truth and the retrieval challenge is to find the other chord sequences representing the same song. Although the dataset was automatically filtered to exclude identical or erroneous pieces, it still includes many songs that are harmonically atypical. The reason for this is that the files are user-generated, and contain peculiar and unfinished pieces, wrong key assignments, and other errors; it can therefore be considered a "real life" dataset. The chord sequence corpus is available to the research community on request.

We analyse the rankings obtained from the compared similarity measures by calculating the Mean Average Precision (MAP). The MAP is the average precision averaged over all queries, and is a single-figure measure between 0 and 1 [11, Chap. 8, p. 160]. We tested whether the differences in MAP are significant by performing a non-parametric Friedman test with a significance level of $\alpha = 0.05$. We chose the Friedman test because the underlying distribution of the data is unknown, and, in contrast to an ANOVA, the Friedman does not assume a specific distribution of variance. [4] To determine which pairs of measurements differ significantly we conducted a post-hoc Tukey HSD test. This way of significance testing is standard in MIREX.

The MAP scores are displayed in Table 1. There are significant differences between the runs, $\chi^2(3, N = 1775) = 350$, $p < 0.0001$ and also all pairwise differences are statistically significant. Hence, we can conclude that HarmTrace significantly outperforms the other similarity measures, and that using the harmonic information obtained by our model improves similarity estimation on this dataset.

6. DISCUSSION

The results show that using information about the function of a chord improves harmonic similarity. However, not all harmony annotations appeared to be beneficial: although in our experiments the functional categories (*Ton*, *Dom*, *Sub*)

[4] All statistical tests were performed in Matlab 2009a.

did not have a negative effect on the similarity estimation, they did not improve the harmonic similarity either. Perhaps the categories are not distinctive enough to be advantageous. We noticed that similarity measures that did not easily classify chords as similar performed best.

The retrieval task of Section 5 is a difficult one because the song class sizes are very small. Often there is only one related piece in the corpus, and finding it based on its harmony alone is challenging. We believe that this is a sound way of evaluating of harmonic similarity, since nothing else could have influence the results but the chords available in the data. Nevertheless, it is stimulating to think about other ways of evaluating harmonic similarity that go beyond the concept of a cover-song. A fundamental problem is that currently there is no good ground-truth that actually captures the harmonic similarity on a gradual (non-binary) scale. But how should such a ground-truth be established: by performing a large scale user study, or by consulting musical experts? These questions remain unanswered, and pose challenges for future MIR research.

Besides similarity estimation, a model of tonal harmony might be useful for answering other MIR-related questions. For instance, chord labelling or optical music recognition systems often recognise chords from audio or score data. Our model could be used to suggest harmonically-fitting solutions when there is high uncertainty in the data. Another potential application of HARMTRACE would be in the generation of harmonically well-formed chord sequences for software such as Band-in-a-Box. The TPSD and CSAS do not offer such benefits.

The many possible applications of harmony models, like the one in HARMTRACE, together with its positive results in retrieval performance, make us believe that formalising tonal harmony is crucial in understanding the true nature of musical harmony and harmonic similarity.

Acknowledgements This work has been partially funded by the Dutch ICES/KIS III Bsik project MultimediaN, and by the Portuguese Foundation for Science and Technology (FCT) via the SFRH/BD/35999/2007 grant. We thank Martin Rohrmeier for all fruitful discussions, Anja Volk for her helpful comments on an earlier draft, and the anonymous reviewers for their constructive comments and suggestions.

7. REFERENCES

[1] P. Gannon. Band-in-a-Box. PG Music, 1990.

[2] W. B. de Haas, J. P. Magalhães, F. Wiering, and R. C. Veltkamp. HarmTrace: Automatic functional harmonic analysis. Technical Report UU-CS-2011-023, Department of Information and Computing Sciences, Utrecht University, 2011.

[3] W. B. de Haas, M. Robine, P. Hanna, R. C. Veltkamp, and F. Wiering. Comparing harmonic similarity measures. In *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval*, pages 299–315, June 2010.

[4] W. B. de Haas, M. Rohrmeier, R. C. Veltkamp, and F. Wiering. Modeling harmonic similarity using a generative grammar of tonal harmony. In *Proceedings of ISMIR*, 2009.

[5] W. B. de Haas, R. C. Veltkamp, and F. Wiering. Tonal pitch step distance: A similarity measure for chord progressions. In *Proceedings of ISMIR*, pages 51–56, 2008.

[6] P. Hanna, M. Robine, and T. Rocher. An alignment based system for chord sequence retrieval. In *Proceedings of the 2009 Joint International Conference on Digital Libraries*, pages 101–104. ACM New York, NY, USA, 2009.

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

[8] F. Lerdahl. *Tonal Pitch Space*. Oxford University Press, 2001.

[9] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1996.

[10] J. P. Magalhães and W. B. de Haas. Functional Modelling of Musical Harmony—an Experience Report. In *Proceedings of the 16th ACM SIGPLAN International Conference on Functional Programming*, 2011.

[11] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

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

[13] J. Pickens and T. Crawford. Harmonic models for polyphonic music retrieval. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 430–437. ACM New York, NY, USA, 2002.

[14] H. Riemann. *Vereinfachte Harmonielehre; oder, die Lehre von den tonalen Funktionen der Akkorde*. Augener, 1893.

[15] C. Roads. Grammars as representations for music. *Computer Music Journal*, 3(1):48–55, 1979.

[16] M. Rohrmeier. A generative grammar approach to diatonic harmonic structure. In Anagnostopoulou Georgaki, Kouroupetroglou, editor, *Proceedings of the 4th Sound and Music Computing Conference*, pages 97–100, 2007.

[17] M. Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1), 2011.

[18] H. Schenker. Der Freie Satz. Neue musikalische Theorien und Phantasien, 1935.

[19] T. F. Smith and M. S. Waterman. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[20] M. J. Steedman. A Generative Grammar for Jazz Chord Sequences. *Music Perception*, 2(1):52–77, 1984.

[21] S. D. Swierstra. *Combinator Parsing: A Short Tutorial*, pages 252–300. Springer-Verlag, 2009.

[22] D. Temperley. *The cognition of basic musical structures*. Cambridge, MA, MIT Press, 2001.