VISUALIZING CHORD RECOGNITION PERFORMANCE

Christopher Liscio[1] **Blake Vanberlo**[1] **Aniruddhan Murali**[1]
Shuhui Zhu[1] **Dan Brown**[1]
[1] David R. Cheriton School of Computer Science, University of Waterloo, Canada
clliscio@uwaterloo.ca, dan.brown@uwaterloo.ca

ABSTRACT

We created a visualization tool that helps Automatic Chord Recognition (ACR) developers to characterize system performance across a test data set. Our system's design uses Information Visualization (InfoVis) principles to communicate accuracy more effectively than a table of mean metric scores. We share some of the insights we developed while building our tool, and hope our findings may help inform the design of figures used in future publications, and affect how future ACR system designers improve and present their systems.

1. INTRODUCTION & BACKGROUND

The ACR task has inspired more than 20 years of active research [1], including the development of accuracy measures [2,3]. Researchers use tables of mean scores across a collection of songs to communicate how well their systems perform, but these fail to provide a complete picture.

For a slightly more detailed representation of system performance, researchers often report more than one mean accuracy value, calculated using a segment-based weighted chord symbol recall calculation using standard matching functions [2].

Each score uses increasingly specific matching criteria to convey how well the ACR system can distinguish between different chord qualities: the *root* score only considers whether two chords share a common root note, while the more discriminative *triads* matching function accepts two chords only if the first three voices are common to both of them. Poor results on *root* indicate fundamental concerns about an ACR system's methods, while in an ACR system that can only recognize major and minor chords, the *triads* metric should be expected to report lower values than one that claims to identify a larger vocabulary.

Researchers have attempted to improve upon the reporting of mean scores, such as McFee and Bello's use of box plots to help convey the relative performance of different system architectures [4]. Some researchers include more sophisticated figures to help them explain individual song results [3–5]. Unfortunately, most examples from the literature stray from the best practices suggested by the field of InfoVis [6].

For example, Kinnaird and McFee included a plot [3, Figure 9, center] for a song with a low `triads` score. The predominant error in this song is that the estimates failed to identify the *power chords* (e.g. `E:maj(*3)` vs `E:maj`) in the reference. Unfortunately, this plot makes it difficult for the viewer to understand *just how close* these estimates are to the annotation. There is little help for the reader to make this visual connection between the two chords: one could use color, proximity, or both to indicate the close relationship between these chords.

These shortcomings in the literature motivated us to build a visualization system to help researchers find songs with error patterns like these, and to quickly gauge the severity of missed estimates as a tool in either better understanding the limitations of their system or improving it.

2. SYSTEM INPUT

Conceptually, our system requires the following for each song in the test set: (1) identifying information (i.e. metadata), (2) the list of intervals with ground truth chord annotations, and (3) the ACR system's output intervals with their estimated chord labels. We import this data in a local database, where we also store pre-calculated results from the `mir_eval` package [7]. For example, we merge time intervals from each song's reference and estimate lists, including their respective chord labels. Each of these merged interval rows also includes a column that stores the result of comparing the ground truth and estimate using the *root* criteria, one for the *thirds* criteria, and so on.

This arrangement of data provided us with a great deal of flexibility. For example, we can calculate vocabulary statistics using SQL queries, and also use queries to provide data for the visualizations discussed below.

3. METRIC SCORE OVERVIEWS

Following the convention in the literature, we present overview scores for a number of the standard metrics. However, instead of reporting only mean values, we use box plots [8] with outliers to represent the distribution of scores across the entire test data set.

By explicitly rendering outliers, we offer a simple mechanism for users to investigate those songs further. When users hover their mouse over an outlier score, they

Late-Breaking / Demo Session Extended Abstract, ISMIR 2022 Conference

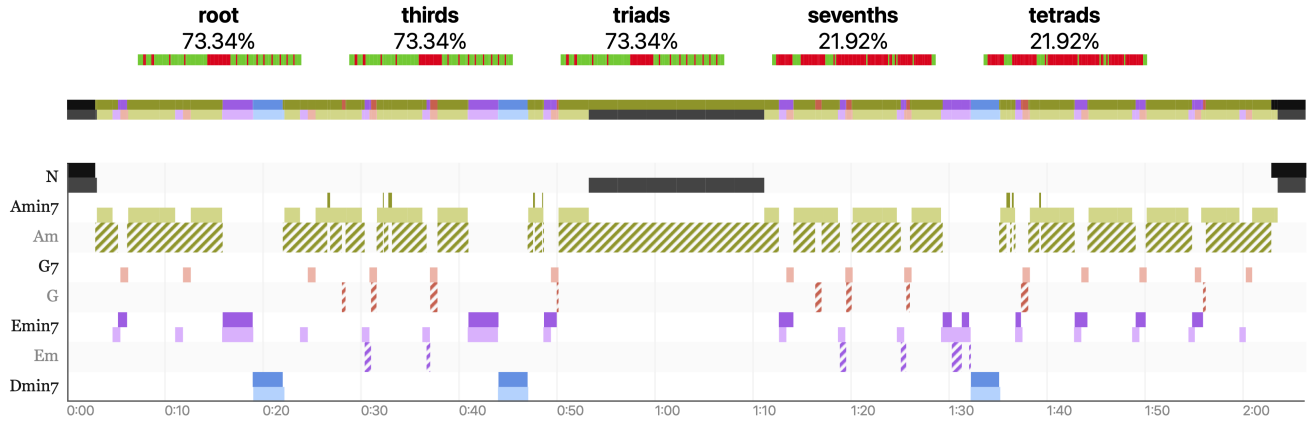

Figure 1. A crop from our song results screen, showing both our standard score pictograms, and the timeline browser. Note the mis-identification of `A:min7` as `A:min`, which does not occur anywhere in the ground truth.

can immediately see which song it came from. Clicking the outlier reveals a more detailed presentation of the differences between the ACR system's estimates and the ground truth annotations.

4. INDIVIDUAL SONG RESULTS

4.1 Standard Metric Pictograms

In each song's detail view, we list identifying information and standard numeric metric scores at the top of the screen, as they are most salient for ACR researchers. However, we improve upon these static values by including a novel *pictogram* to illustrate the distribution of matching failures over the song's duration. At a glance, this visual representation of the matching function's output supplies two key pieces of information: (1) the *goodness* of the score (i.e. predominantly green, or red), and (2) the frequency of errors across the duration of the song.

For example, in Figure 2 we show pictograms for a score of 90% that appears with a predominantly green fill color, but the contribution from the remaining 10% differs greatly.

Figure 2. Three standard metric pictograms, each depicting the same accuracy score of 90%.

4.2 Timeline Browser

When it is deemed necessary to investigate the cause of matching failures, our timeline browser provides researchers with a rich, interactive display that should feel somewhat familiar. For example, we use a standard arrangement that displays chord labels on the vertical, and time on the horizontal axis. However, we break from tradition in a few key ways:

Chords are *simplified* and *sorted* on the vertical axis. For example, we treat `C#:maj`, `C#:(1,3,5)`, and `Db` as identical labels on the timeline. We then sort the labels first by their root note, then by qualities according to our own ordering scheme.

Chords are assigned colors from a fixed palette according to their root note. We optimized our color palette to maintain an even perceptual distance between the hues of adjacent pitch classes, and to contrast well against a white background.

Ground truth intervals appear directly below the estimates. By pairing these elements within a shared *lane* for each of the labels, it is made clear when the two are not in agreement.

A pictogram overview of the estimated and reference annotations. Like the standard metric pictograms, these are a compact representation of estimation results over the song's timeline.

Out-of-vocabulary chord labels appear distinctly, in their correct vertical position. Using a striped fill pattern that occupies the full lane height, intervals that are assigned labels outside the vocabulary of chords in the ground truth stand out clearly.

5. FUTURE WORK

We plan to share more details about our system in a future publication: the derivation of our chord sorting and color choices, as well as concrete examples that highlight the effectiveness of our tool.

6. ACKNOWLEDGMENTS

The authors wish to thank Brian McFee of the Music and Audio Research Lab (MARL) at NYU Steinhardt for supplying the ground truth and estimates that we used to demonstrate the visualization system.

7. REFERENCES

[1] J. Pauwels and G. Peeters, "Evaluating automatically estimated chord sequences," in *2013 IEEE Interna-*

tional Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.

[2] C. Harte, "Towards automatic extraction of harmony information from music signals," Ph.D. dissertation, 2010.

[3] K. M. Kinnaird and B. McFee, "Automatic hierarchy expansion for improved structure and chord evaluation," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 81–92, 2021.

[4] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition." in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, 2017, pp. 188–194.

[5] D. Odekerken, H. V. Koops, and A. Volk, "Improving audio chord estimation by alignment and integration of crowd-sourced symbolic music," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 141–155, 2021.

[6] T. Munzner, *Visualization analysis and design*. CRC press, 2014.

[7] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common MIR metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.

[8] K. Potter, H. Hagen, A. Kerren, and P. Dannenmann, "Methods for presenting statistical information: The box plot." in *VLUDS*. Citeseer, 2006, pp. 97–106.