AN INTERACTIVE WORKFLOW FOR GENERATING CHORD LABELS FOR HOMORHYTHMIC MUSIC IN SYMBOLIC FORMATS

Yaolong Ju[1] **Samuel Howes**[1] **Cory McKay**[2]
Nathaniel Condit-Schultz[3] **Jorge Calvo-Zaragoza**[4] **Ichiro Fujinaga**[1]
[1] Schulich School of Music, McGill University, Canada
[2] Department of Liberal and Creative Arts, Marianopolis College, Canada
[3] School of Music, Georgia Institute of Technology, USA
[4] Department of Software and Computing Systems, University of Alicante, Spain
yaolong.ju@mail.mcgill.ca, samuel.howes@mail.mcgill.ca, cory.mckay@mail.mcgill.ca,
natcs@gatech.edu, jcalvo@dlsi.ua.es, ichiro.fujinaga@mcgill.ca

ABSTRACT

Automatic harmonic analysis is challenging: rule-based models cannot account for every possible edge case, and manual annotation is expensive and sometimes inconsistent, undermining the training and evaluation of machine learning models. We present an interactive workflow to address these problems, and test it on Bach chorales. First, a rule-based model was used to generate preliminary, consistent chord labels in order to pre-train three machine learning models. These four models were grouped into an ensemble that generated chord labels by voting, achieving 91.4% accuracy on a reserved test set. A domain expert then corrected only those chords that the ensemble did not agree on unanimously (20.9% of the generated labels). Finally, we used these corrected annotations to re-train the machine learning models, and the resulting ensemble attained an accuracy of 93.5% on the reserved test set, a 24.4% reduction in the number of errors. This versatile interactive workflow can either work in a fully automatic way, or can capitalize on relatively minimal human involvement to generate higher-quality chord labels. It combines the consistency of rule-based models with the nuance of manual analysis to generate relatively inexpensive high-quality ground truth for training effective machine learning models.

1. INTRODUCTION AND BASIC METHODOLOGY

In general, harmonic analysis refers to the identification of harmonies from the musical surface. As a key part of the foundation of modern Western music theory, harmonic analysis is inherently complex. It is based on low-level sensory distinctions (consonance vs dissonance), local constructs (counterpoint, voice-leading), and global musical structures (harmonic function, form, tonality, etc.). Learning it is thus a time-consuming process, requiring years of training. Furthermore, many prominent music theorists (e.g., Rameau, Riemann, Schenker) have proposed different approaches to harmonic analysis. This means it is often possible to analyze the same passage in numerous legitimate ways. For example, some analysts prefer interpretations with fewer chords, while others prefer interpretations with more frequent harmonic changes. We characterize these general strategies as "melodic" and "harmonic", respectively (Fig. 1 illustrates these interpretive strategies). Complicating matters further, analysts often disagree, and are not always internally consistent [12]. Given the complexity, subjectivity, and inconsistency of harmonic analysis, it is challenging to systemize it.

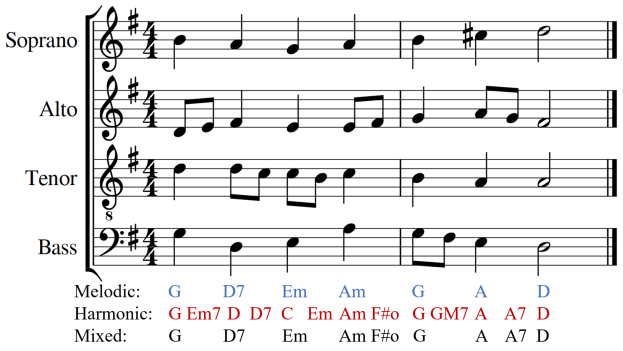

Figure 1. A passage with important differences between melody-oriented (blue) and harmony-oriented (red) analyses. The final analysis (black) mixes the two styles. Such inconsistencies are quite common, even between expert analyses.

In spite of these challenges, there have been various attempts to automate harmonic analysis. Data generated by automated approaches could be used to populate a large-scale, searchable database, which would serve as an invaluable resource for music research. For example, such a database could be used in corpus studies to answer re-

search questions about musical style or the development of modern harmonic practices. Automatic harmonic analysis can also be used in automatic composition and interactive accompaniment systems.

Some researchers have developed rule-based (RB) models for automatic harmonic analysis [4, 8, 10, 21–23]. Although these approaches generate chord labels that are internally consistent, they often fail to produce correct analyses for even moderately exceptional passages, as it is extremely complicated to define rules that are comprehensive enough to account for all possibilities.

Other researchers have made use of manual annotations by experts, who can better respond to exceptions [2,5,6,9,16,17]. Such ground truth can be used to train machine learning (ML) models for automatic harmonic analysis [3, 11, 14, 15, 18, 20, 24]. Although the annotations created by human analysts are more nuanced, manual harmonic annotations require an enormous amount of time and expertise, and can be inconsistent [12], which may undermine a ML model's effectiveness, especially when limited amounts of training data are available.

Due to these difficulties, few large high-quality datasets and automatic harmonic analysis models exist, a situation that has significantly limited the computational study of Western harmony.

In this paper, we combine the strengths of existing approaches to address the common problems of automatic harmonic analysis within a single interactive workflow, using a set of largely homorhythmic [1] Bach chorales. The proposed workflow is illustrated in Fig. 2 and described below:

1. To solve the problem of analytical inconsistency, we use an existing RB model [4] to generate preliminary, consistent chord labels according to a particular analytical style.

2. These analyses are used to pre-train three ML models, [2] which together with the RB model form an algorithm ensemble, where each model within the ensemble labels all the chords. The most-preferred chord labels [3] are then output as *Analysis 1*.

3. To improve the quality of the analyses, a human expert examines only those chords for which the ensemble did not agree unanimously, and corrects them as needed. We call this process "partial manual modification". Compared to annotating chorales from scratch, the amount of required work for the expert is significantly reduced. The first three steps of this workflow are shown in Part 1 of Fig. 2.

4. Once the expert's corrections are obtained (*Analysis 2*), we re-train the ML models. The most-preferred chord labels from the new ensemble are chosen as the final chord labels (*Analysis 3*), which is shown in Part 2 of Fig. 2. This paradigm of manually modifying the generated data and re-training the ML models is known as "interactive machine learning" [1,7].

This workflow is not limited to Bach chorales. With an adapted RB model (Model 4 in Fig. 2), it can easily be applied to other genres of music in a fully automatic way (ending with Analysis 1) or interactively if an expert analyst is available (ending with Analysis 3). The source code, data, and results from this project can be found at: https://bit.ly/2QUdGwH.

2. DETAILS OF METHODOLOGY

This section introduces additional details of the interactive workflow shown in Fig. 2 and described in Section 1.

2.1 Input Data Encoding and Processing

The workflow currently accepts music encoded in Humdrum's **kern symbolic representation. Any other formats that can be faithfully converted to **kern can also be used.

Each chord label consists of the letter-name of the root and the quality of the chord (e.g., C major). Triads can be major, minor, or diminished; and seventh chords can be major, minor, dominant, half-diminished, or fully diminished. Functional Roman numerals are not used, and chordal inversions are not specified.

Chord labels are appended to the original **kern file for each chorale and aligned with the music as "onset slices" [11,13], as shown in Fig. 4. An onset slice is formed whenever a new note onset occurs in *any* musical voice, and consists of a list of all pitch classes sounding at that moment.

Additionally, all chorales and corresponding chord labels were transposed to the same key to make the tonal relationships between pitch classes consistent across the dataset. [4]

2.2 Input Features

Each onset slice is mapped to a feature vector for processing by Model 1, Model 2, and Model 3 of the workflow. These features, [5] and the codes used to refer to them in Section 3, are as follows:

1. **PC12** : A 12-D binary vector of enharmonic pitch classes present in the slice.

2. **M**: A 3-D indication of the metrical context of the slice (down-beat, on-beat, off-beat).

3. **O**: A 12-D vector indicating which PC12 pitch classes are real onsets and which are artificial (see Fig. 4).

4. **Wn**: A variable size vector containing the (non-Wn) features from the n previous and following slices

[1] Homorhythm is a texture where all parts share a very similar rhythm, as in Fig. 1. It is commonly used in hymn and chorale settings.

[2] See the caption of Fig. 2 for the details of these models.

[3] If there is a tie, prefer the label for which the rule-based algorithm voted.

[4] The built-in key transposition function from music21 was used, with the Aarden-Essen key profile (https://bit.ly/2FSIwQY). Chorales were transposed to C major or A minor depending on their mode.

[5] A multi-label one-hot schema was used to encode the features as inputs for the ML algorithms.

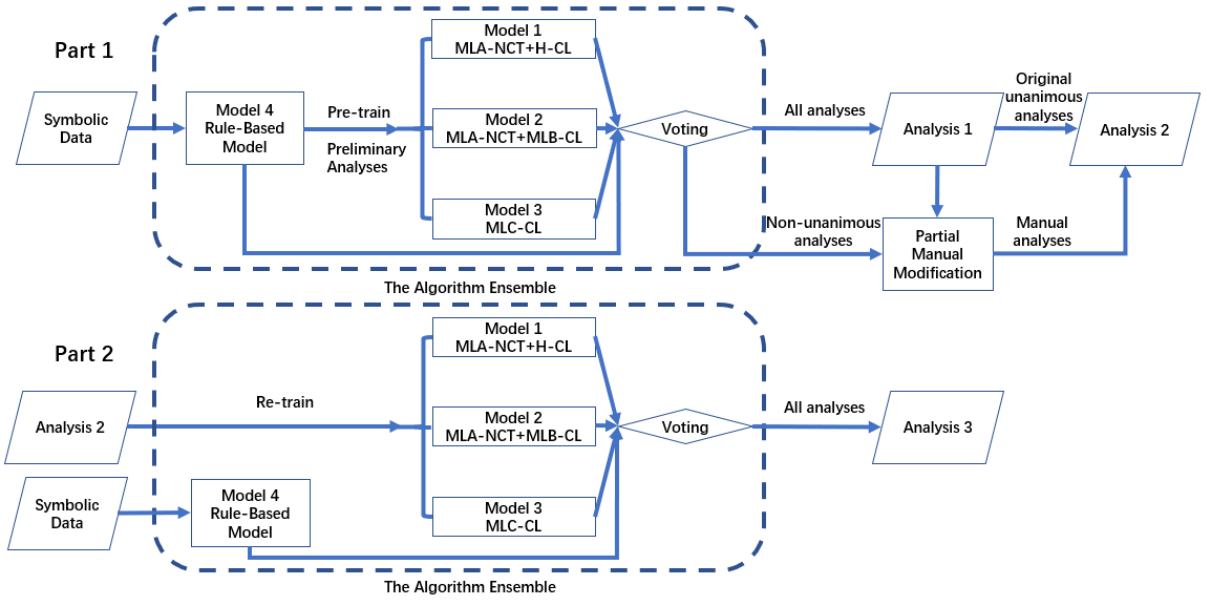

Figure 2. Interactive workflow for automatic harmonic analysis. There are four models within the algorithm ensemble, three of which are trainable. Models 1 and 2 both use a machine learning algorithm (MLA) to identify and remove non-chord tones (NCTs). After this, Model 1 (MLA-NCT+H-CL) uses a heuristic (H) algorithm and Model 2 (MLA-NCT+MLB-CL) uses a ML algorithm (MLB) to infer chord labels (CL) from the remaining chord tones. We term this process "NCT-first harmonic analysis", as shown on the right side of Fig. 3. Model 3 (MLC-CL) uses a single ML algorithm (MLC) to infer chord labels (CL) directly from the pitch-class collections, without removing NCTs. We term this process "direct harmonic analysis", as shown on the left side of Fig. 3.

(e.g., W1 indicates that features for the directly preceding and directly following slices are included in the features of the current slice). These surrounding slices are called "contextual windows".

The workflow allows for experimentation with different feature configurations. For example, a "PC12M" configuration indicates a 15-D vector, with O and Wn features omitted. This notation is adopted in Section 3.

2.3 Rule-Based Algorithms

We use an existing RB model [4] to generate preliminary chord labels (Model 4 in Fig. 2). This tool is publicly accessible online. [6] A "harmonic" rather than "melodic" style of analysis is used (see Fig. 1), which prefers more chord changes and fewer non-chord tones (NCTs) [19], and is better-suited to the typical chorale texture. An overview of the specific heuristics of this style can be found at: https://bit.ly/2XCmNVo. We also used a heuristic algorithm (H-CL from Fig. 2) in Model 1 to infer chord labels from remaining chord tones. The details of this algorithm can be found at: https://bit.ly/2MBL0dp.

2.4 Machine Learning Algorithms

As shown in Fig. 2, the workflow includes three ML algorithms (MLA, MLB, and MLC) to pre-train. MLA treats NCT identification as a multi-label problem; the output of MLA is a 12-dimensional vector specifying which pitch classes are both present and identified as NCTs; MLB and MLC treat chord labeling as a multi-class problem; they output similar vectors identifying the predicted chord label among all candidates.

We tested Support Vector Machines (SVMs) and Deep Neural Networks (DNNs) as MLA, MLB, and MLC classifiers. For DNN, we used three hidden layers, each with 300 hidden units. Adaptive Moment Estimation was used as an optimizer, with loss functions of binary cross-entropy for MLA and categorical cross-entropy for MLB and MLC. SVM used a linear kernel function.

3. EXPERIMENTS

3.1 Data

The experiments below were performed on a modified [7] dataset of Bach chorales originally produced by Craig Sapp. [8] This modified dataset consists of 369 chorales.

To evaluate the performance of our workflow, 39 chorales were randomly chosen before the experiments began and partitioned into a set reserved for final testing in Experiment 2. These reserved chorales had their chords hand-labelled in their entirety by a human expert.

The remaining 330 non-reserved chorales were used for training, validation (early-stopping) and internal testing.

[6] https://bit.ly/2Gh6IhA

[7] Available at: https://bit.ly/2VWHB8w. Some corrections were made to the music and Chorale 150 was added to the dataset. Chorales 130 and 316 were excluded, since the original **kern files and the music21-parsed results are different.

[8] https://bit.ly/2D4ju10

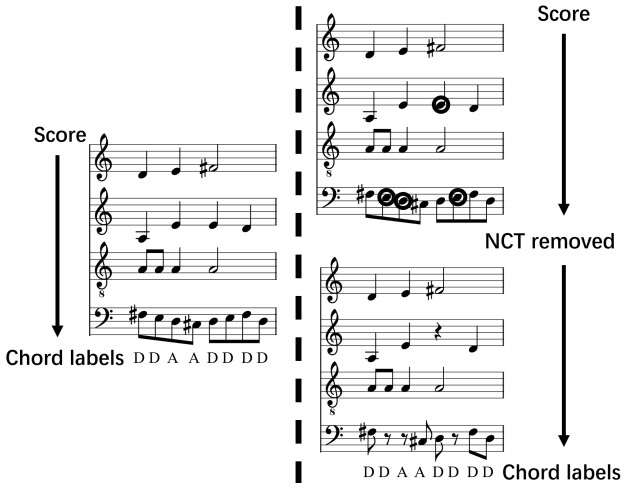

Figure 3. Comparison of "direct harmonic analysis" (left, used by Model 3 in Fig. 2) and "NCT-first harmonic analysis" (right, used by Model 1 and Model 2 in Fig. 2) approaches to automatic harmonic analysis. The former identifies chords directly from the score, while the latter first identifies and removes non-chord tones from the score, and then generates chord labels from the remaining chord tones.

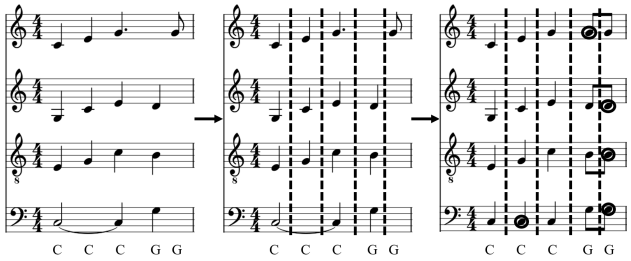

Figure 4. Illustration of note onset slices, aligned with chord labels. An onset slice is created whenever a new note onset occurs in *any* musical voice (middle). Any note sustained from a previous slice becomes an "artificial onset" in the new slice (right, circled).

The initial "ground truth" for these remaining 330 chorales consisted of the labels predicted by the RB model (Model 4), which was found to be quite effective, if not perfect [4]. This imperfect "ground truth" was used in Experiment 1 (see Section 3.2) to get a preliminary sense of how well the workflow's component classifiers performed. Final evaluation was performed in Experiment 2 (see Section 3.3) with the proper, hand-annotated 39-chorale reserved test set.

3.2 Experiment 1

Experiment 1 tested the effectiveness of several different workflow configurations by experimenting on varying input features and learning algorithms (see Section 2.4). The performance of Models 1, 2, and 3 from Fig. 2 were tested.

3.2.1 Experimental Setup

Ten-fold cross-validation was performed on the 330 non-reserved chorales described in Section 3.1. For the DNN experiments, we divided the non-reserved portion of the dataset (330 chorales) into training (80%), validation (10%) and internal testing (10%) folds. The SVM data was divided into training (90%, the union of the DNN training and validation sets) and internal testing (10%, matching the DNN internal test sets) folds. When the W features were included (see Section 2.2), n was set to 1 for MLA and MLC, and to 2 for MLB (represented as W1/2).

3.2.2 Results

The results of Experiment 1 are shown in Table 1. The highest classification value of 90.1% was achieved by Model 2 using PC12MOW1/2 input features. Results show that the addition of a small contextual window (feature Wn) improved the performances of Model 2 and Model 3 significantly.[9] This reflects the general music theoretical understanding that, in cases of ambiguous harmony (e.g., an incomplete chord), a chord's immediate context is essential to label it properly.

It is important to note that these Experiment 1 findings are based on imperfect ground truth (see Section 3.1), and so must be interpreted more as preliminary indications rather than as confirmed truth. Experiment 2 was performed in order to obtain more empirically meaningful results.

3.3 Experiment 2

Experiment 2 compared the performance of the classifier ensemble after fully automated training (Analysis 1 in Fig. 2) with that of the ensemble after human-assisted re-training (Analysis 3 in Fig. 2). This set of experiments involved evaluation on a reserved expert-labelled test set (see Section 3.1).

3.3.1 Experimental Setup

Classification models were pre-trained, had their outputs manually corrected, re-trained, and tested using the full workflow described in Section 1. Pre-training was done using the Model 4 output, just as in Experiment 1.

For the DNN training, we used 90% of the 330 non-reserved chorales as the training set and 10% as the validation set. A cross-validation-like training scheme was used: we conducted 10 experiments by training 10 models with rotated training and validation folds, while the testing fold (39 reserved chorales) remained the same. All 330 non-reserved chorales were used to train each of the SVM classifiers. Only the PC12MOW1/2 input features (see Section 2.2) were used in Experiment 2. For the W features, n was set to 1 for MLA and MLC, and to 2 for MLB (represented as W1/2).

Once Analysis 1 (see Fig. 2) was obtained, the human expert manually corrected only those chords that the ensemble did not agree on unanimously. The corrected labels (Analysis 2) were then used to re-train Models 1, 2, and 3. The 39 manually-labelled reserved test chorales were then used to test the original pre-trained models, and then the

[9] $p<0.05$ in Students' t-tests comparing all Model 2 and 3 accuracies for PC12 and PC12M with those of PC12W1/2 and PC12MW1/2.

Model	Metric	PC12	PC12M	PC12W1/2	PC12MW1/2	PC12MOW1/2
SVM	CA1	**81.7±1.4%**	81.6±1.4%	82.7±1.0%	83.0±1.0%	83.5±0.9%
	CA2	73.0±1.5%	73.1±1.6%	85.4±1.3%	86.1±1.5%	87.4±1.5%
	CA3	74.9±1.6%	75.6±1.5%	85.4±1.3%	85.9±1.3%	87.7±1.5%
DNN	CA1	81.0±1.5%	**81.7±1.5%**	85.3±0.9%	85.6±0.9%	85.8±0.9%
	CA2	74.2±1.8%	75.1±1.6%	**88.5±1.3%**	**89.6±1.3%**	**90.1±1.5%**
	CA3	74.6±1.8%	75.3±1.4%	87.5±1.7%	88.3±1.7%	89.0±2.0%

Table 1. Experiment 1 cross-validation classification accuracies, averaged across folds. Uncertainty values indicate standard error across folds. Values indicate the percentage of onset slices "correctly" classified by Model 1 (CA1), Model 2 (CA2), and Model 3 (CA3), based on the Model 4 "ground truth". Columns indicate features (see Section 2.2) and rows indicate machine learning algorithms (see Section 2.4). The best performance in each column is highlighted in bold.

Model	Metric	PC12MOW1/2 Pre-trained	PC12MOW1/2 Re-trained
SVM	CA1	85.9%	87.0%
	CA2	88.6%	89.8%
	CA3	87.7%	89.3%
	CAVote	**91.4%**	92.7%
	PUA	79.1%	79.0%
DNN	CA1	85.4±0.2%	88.1±0.2%
	CA2	88.9±0.3%	91.3±0.4%
	CA3	87.9±0.7%	90.5±0.3%
	CAVote	90.9±0.2%	**93.5±0.2%**
	PUA	80.4±1.2%	79.7±0.4%
RB	CA4	90.7%	

Table 2. Experiment 2 classification accuracies on the reserved test set. DNN values are averaged across models trained using different training/validation sets, and uncertainty values indicate standard error across these folds. Values indicate how many onset slices were correctly classified by Model 1 (CA1), Model 2 (CA2), Model 3 (CA3), Model 4 (CA4), the ensemble as a whole (CAVote), and just those CAVote predictions that were unanimous (PUA). "PC12MOW1/2" indicates the input features (see Section 2.2. "Pre-trained" indicates performance before manual correction (i.e., Analysis 1 in Fig. 2), and "Re-trained" indicates performance after re-training on the corrected data (i.e., Analysis 3 in Fig. 2). The best performance in each column is highlighted in bold.

re-trained models. Performance on this reserved test set is shown in Table 2.

3.3.2 Results

One can see in Table 2 that the original RB algorithm (Model 4 in Fig. 2) attains a chord accuracy of 90.7%, which serves as our baseline. The highest accuracy obtained by the pre-trained ensemble is 91.4%, using PC12MOW1/2, SVM classifiers, and voting. This (pre-trained) performance is achieved without any expert human intervention. It is of interest that CAVote here is higher than CA4, even though the classifiers in CAVote were trained on the RB output; this is perhaps because the RB model is overfitting the theoretical model underlying it, and that the pre-trained ensemble trained on it may in fact be smoothing out some of this overfitting to result in a slightly more general model. A comparison of Table 1 and Table 2 indicates that the Table 1 performance with artificial ground truth is quite similar to the performance of Table 2 pre-trained classifiers on the proper test set; this encouragingly suggests that there is little or no overfitting.

Table 2 also shows that performance improved after re-training in most cases. [10] The best-performing [11] configuration attains an accuracy of 93.5%, using voting DNNs trained on PC12MOW1/2 features.

The partial manual modification workflow is also found to be relatively efficient, as the expert analyst is only required to provide manual analyses for about 20.9% [12] of all slices. Compared to examining and annotating every slice, the amount of required work is reduced substantially.

4. DISCUSSION

According to the results, our interactive workflow performed well on the Bach dataset using a "harmonic" style of analysis. It was found that quite good performance could be achieved with our rule-based model (90.7% on the reserved test data), that performance could be improved slightly using the RB model to self-train a classifier ensemble (91.4% on the test data), and that still greater improvements resulted from partial manual modification and re-training (93.5% on the test data). Although these improvements may seem small in absolute terms, they are statistically significant, and they represent meaningful fractional decreases in the error rate (drops of 7.5% comparing pre-trained CAVote to RB, 30.1% comparing re-trained CAVote to RB, and 24.4% comparing re-trained CAVote to pre-trained CAVote). Of particular importance, the first two approaches require no human intervention, and the third requires much less expert labor than full manual annotation.

Figure 5 provides an illustration of how this approach can be effective, using an excerpt from one of the test set chorales. Although some algorithms within the ensemble

[10] $p<0.05$ in Students' t-tests comparing results before and after re-training for CA1, CA2, CA3, and CAVote, but not PUA.

[11] $p<0.05$ in Students' t-tests comparing results of CAVote to CA1, CA2, CA3, and CA4.

[12] This value is inferred from Table 2: 100% - PUA.

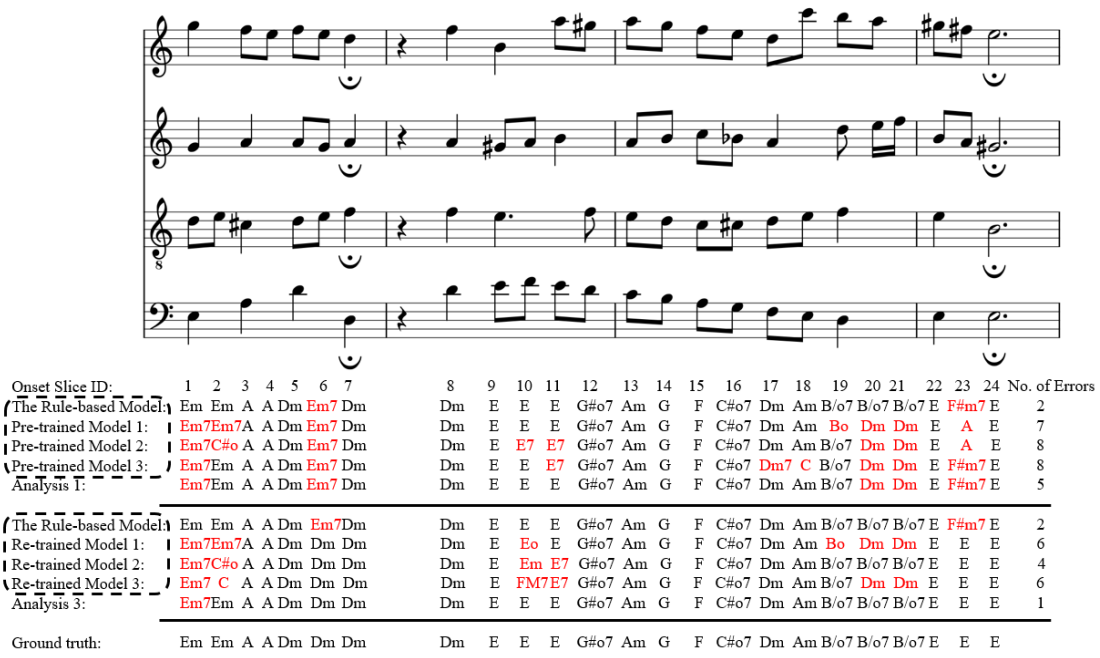

Figure 5. An illustration of how classifications evolve as processes proceed as outlined in Fig. 2, based on measures 9 through 12 of BWV 315 "Gib dich zufrieden und sei stille". Chord labels were generated by a DNN-based algorithm ensemble using PC12MOW1/2 features (see Section 2.2). The algorithm ensemble is made up of the four models within the dashed rectangle, which vote to generate Analyses 1 and 3. The labels above the first horizontal line were generated in a fully automatic way, without any human intervention. The labels between the two horizontal lines (other than the rule-based model) were generated automatically after re-training on partially corrected data. The chord labels highlighted in red are errors compared to the ground truth provided by an expert analyst.

make errors, the re-trained ensemble ultimately generates better answers in Analysis 3. Upon examining the errors, we find that some of them are reasonable alternative versions of the ground truth: chords with the same roots, but with or without an added seventh (slices 1, 11, 17, and 19); or chords that are subsets of the ground truth chords (slices 20 and 21). As a result, some of the "errors" that the ensemble makes in this particular excerpt are in fact theoretically acceptable answers. This is encouraging, as it suggests that at least some of the "mistakes" made by the classifiers may not in fact be mistakes at all. We still count them as mistakes, however, because consistency in analytical style is one of the goals of this work.

5. CONCLUSION AND FUTURE RESEARCH

We present a versatile interactive workflow for generating chord labels for homorhythmic music. It can be used in a fully automatic way or, with a relatively small amount of effort from an expert human analyst who corrects a small, automatically selected fraction of the generated analyses, a re-trained classifier ensemble can be produced that performs even better.

Results show that this workflow is quite compelling: it combines the consistency of rule-based models with the nuance of manual analysis to generate relatively inexpensive, high-quality ground truth for training effective machine learning models. The resulting classifier ensemble is able to automatically generate highly consistent and accurate chord labels, which can serve as invaluable resources for musicians, composers, and music researchers alike.

There are currently a few limitations to our research. First, music21's automatic key-finding might not be ideal for our dataset (early tonal music), and may have resulted in reduced performance due to faulty transpositions. Instead of transposing all chorales to the same key, a better, but more complicated solution would be to augment our data by transposing all chorales to all 12 possible keys. Second, the RB model can be improved to include chords of other qualities (e.g., augmented-sixth chords). Finally, the ground-truth annotations were prepared by a single expert annotator, and it would be better to repeat this process using annotations from multiple experts.

An important next step will be to test this workflow using other analytical styles (e.g., the "melodic" style), which can be done simply by specifying different heuristics in the RB model. We also plan to tackle the larger category of homophonic music, which includes any music with a primary melodic line accompanied by harmonic support. A greater variety of homophonic textures poses a challenge to our RB model because more individual onset slices are harmonically ambiguous, requiring larger contextual windows to correctly interpret the harmony. In light of this, we will modify our workflow to address homophonic music accordingly. Finally, we will investigate training and evaluation protocols that permit multiple valid chord labels per slice.

ACKNOWLEDGEMENT

We would like to thank the Social Sciences and Humanities Research Council of Canada (SSHRC) and the Fonds de recherche du Québec-Société et culture (FRQSC) for their generous funding. We would also like to acknowledge the contributions of our many collaborators on the Single Interface for Music Score Searching and Analysis (SIMSSA) project, especially Emily Hopkins and Gabriel Vigliensoni.

6. REFERENCES

[1] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[2] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An expert ground truth set for audio chord recognition and music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 633–638, 2011.

[3] Tsung-Ping Chen and Li Su. Functional harmony recognition of symbolic music data with multi-task recurrent neural networks. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 90–97, 2018.

[4] Nathaniel Condit-Schultz, Yaolong Ju, and Ichiro Fujinaga. A flexible approach to automated harmonic analysis: Multiple annotations of chorales by Bach and Prætorius. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 66–73, 2018.

[5] Trevor de Clercq and David Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(1):47–70, January 2011.

[6] Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 728–734, 2015.

[7] Jerry Alan Fails and Dan R. Olsen Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 39–45, 2003.

[8] Mark Granroth-Wilding. *Harmonic analysis of music using combinatory categorial grammar*. PhD thesis, The University of Edinburgh, 2013.

[9] Thomas Hedges and Martin Rohrmeier. Exploring Rameau and beyond: A corpus study of root progression theories. In *Proceedings of the 1st International Conference on Mathematics and Computation in Music*, pages 334–337, 2011.

[10] Tim Hoffman and William P. Birmingham. A constraint satisfaction approach to tonal harmonic analysis. Technical report, *Electrical Engineering and Computer Science Department. CSE-TR-397-99. University of Michigan*, 2000.

[11] Yaolong Ju, Nathaniel Condit-Schultz, Claire Arthur, and Ichiro Fujinaga. Non-chord tone identification using deep neural networks. In *Proceedings of the 4th International Workshop on Digital Libraries for Musicology*, pages 13–16, 2017.

[12] Hendrik Vincent Koops, W. Bas de Haas, John Ashley Burgoyne, Jeroen Bransen, Anna Kent-Muller, and Anja Volk. Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research*, 48:1–21, 2019.

[13] Pedro Kröger, Alexandre Passos, Marcos Sampaio, and Givaldo De Cidra. Rameau: A system for automatic harmonic analysis. In *Proceedings of International Computer Music Conference*, pages 273–281, 2008.

[14] Kristen Masada and Razvan Bunescu. Chord recognition in symbolic music using semi-Markov conditional random fields. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, pages 272–278, 2017.

[15] Lesley Mearns. *The computational analysis of harmony in Western art music.* PhD thesis, Queen Mary University of London, 2013.

[16] Néstor Nápoles López. *Automatic harmonic analysis of classical string quartets from symbolic score*. Master's thesis, Universitat Pompeu Fabra, 2017.

[17] Markus Neuwirth, Daniel Harasim, Fabian Claude Moss, and Martin Rohrmeier. The annotated Beethoven corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets. *Frontiers in Digital Humanities*, 5:16, 2018.

[18] Alexandre Passos, Marcos Sampaio, Pedro Kröger, and Givaldo De Cidra. Functional harmonic analysis and computational musicology in Rameau. In *Proceedings of the 12th Brazilian Symposium on Computer Music*, pages 207–210, 2009.

[19] Ian Quinn. Are pitch-class profiles really key for key? *Zeitschrift der Gesellschaft für Musiktheorie [Journal of the German-speaking Society of Music Theory]*, 7(2):151–163, 2010.

[20] Christopher Raphael and Joshua Stoddard. Functional harmonic analysis using probabilistic models. *Computer Music Journal*, 28(3):45–52, 2004.

[21] Craig Stuart Sapp. Computational chord-root identification in symbolic musical data: Rationale, methods, and applications. *Computing in Musicology*, 15:99–119, 2007.

[22] Ricardo Scholz, Vitor Dantas, and Geber Ramalho. Automating functional harmonic analysis: The Funchal system. In *Proceedings of the 7th IEEE International Symposium on Multimedia*, pages 759–764, 2005.

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

[24] Wan Shun Vincent Tsui. *Harmonic analysis using neural networks*. Master's thesis, University of Toronto, 2002.