Don't hide in the frames: Note- and pattern-based evaluation of automated melody extraction algorithms

Klaus Frieler*
University of Music "Franz Liszt"
Weimar, Germany

Doğaç Başaran
Audible Magic
London, United Kingdom

Frank Höger
University of Music "Franz Liszt"
Weimar, Germany

Hélène-Camille Crayencour
Centre National de la Recherche Scientifique
Paris, France

Geoffroy Peeters
Télécom ParisTech
Paris, France

Simon Dixon
Center for Digital Music, Queen Mary University of London
London, United Kingdom

ABSTRACT

In this paper, we address how to evaluate and improve the performance of automatic dominant melody extraction systems from a pattern mining perspective with a focus on jazz improvisations. Traditionally, dominant melody extraction systems estimate the melody on the frame-level, but for real-world musicological applications note-level representations are needed. For the evaluation of estimated note tracks, the current frame-wise metrics are not fully suitable and provide at most a first approximation. Furthermore, mining melodic patterns (n-grams) poses another challenge because note-wise errors propagate geometrically with increasing length of the pattern. On the other hand, for certain derived metrics such as pattern commonalities between performers, extraction errors might be less critical if at least qualitative rankings can be reproduced. Finally, while searching for similar patterns in a melody database the number of irrelevant patterns in the result set increases with lower similarity thresholds. For reasons of usability, it would be interesting to know the behavior using imperfect automated melody extractions. We propose three novel evaluation strategies for estimated note-tracks based on three application scenarios: Pattern mining, pattern commonalities, and fuzzy pattern search. We apply the proposed metrics to one general state-of-the-art melody estimation method (Melodia) and to two variants of an algorithm that was optimized for the extraction of jazz solos melodies. A subset of the Weimar Jazz Database with 91 solos was used for evaluation. Results show that the optimized algorithm clearly outperforms the reference algorithm, which quickly degrades and eventually breaks down for longer n-grams. Frame-wise metrics provide indeed an estimate for note-wise metrics, but only for sufficiently good extractions, whereas F1 scores for longer n-grams cannot be predicted from frame-wise F1 scores at all. The ranking of pattern commonalities between performers can be reproduced with the optimized algorithms but not with the reference algorithm. Finally, the size of result sets of pattern similarity searches decreases for automated note extraction and for larger similarity thresholds but the difference levels out for smaller thresholds.

KEYWORDS

automatic melody extraction, evaluation, jazz, pattern mining

ACM Reference Format:
Klaus Frieler, Doğaç Başaran, Frank Höger, Hélène-Camille Crayencour, Geoffroy Peeters, and Simon Dixon. 2019. Don't hide in the frames: Note- and pattern-based evaluation of automated melody extraction algorithms. In *6th International Conference on Digital Libraries for Musicology (DLfM '19), November 9, 2019, The Hague, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3358664.3358672

1 INTRODUCTION

In our ongoing research project "Dig That Lick" (DTL), we are investigating pattern usage in jazz improvisations [4, 16, 17], such as the transmission of patterns between performers over time. The main idea is to create a large database of several thousands of transcriptions of monophonic solos spanning the history of jazz. Since manual transcription is very time-consuming, this goal is attainable (with reasonable effort) only with the help of reliable automatic melody extraction (AME) systems. The quality of the transcriptions is of great importance because of the high standards in the jazz research community and any downstream applications and results derived therefrom might be strongly influenced by it. In order to produce reliable musicological results based on automated transcription, the performance of the system has to be evaluated carefully, optimally for all application scenarios separately, as it is not *a priori* clear, how derived measurements and secondary tasks are affected by transcription errors.

The present paper tries to address some of these issues in the context of a specific research project. Hence, some of the metrics presented herein might not be of much use for other AME applications, but we like to demonstrate by giving examples how application-specific evaluation metrics can be constructed. We want to stress the point that the usual approach in Music Information Retrieval of using standardized evaluation measures for very general algorithms might not sufficiently informative if it comes to very specific applications such as musicological research with high quality standards.

We thus ask the following questions:

*Corresponding author, klaus.frieler@hfm-weimar.de

DLfM '19, November 9, 2019, The Hague, Netherlands

ACM ISBN 978-1-4503-7239-8/19/11…$15.00
https://doi.org/10.1145/3358664.3358672

- How well does an AME system perform in extracting melodic patterns from jazz solos?
- How can an AME system be evaluated best in regard to secondary applications?
- How informative are standardized evaluation measures about AME system performance in specific tasks?

AME systems are normally evaluated using a frame-wise comparison with the ground truth [2, 21], but for most musicological applications the note level is the relevant level, which calls for a note-based evaluation [22]. Moving from frame-wise pitch tracks to note tracks involves one more algorithmic step, e g., a Hidden Markov Model used for smoothing with the Viterbi algorithm [8, 22], which might be a further source for errors, but also an opportunity to compensate for lower-level errors.

However, note-level evaluation might still not be sufficient, if, as in our case, the objects of interest are patterns, i. e., sequences of consecutive notes or n-grams. It is easy to see that note-level errors propagate geometrically for consecutive sequences. If the mean accuracy for transcribing a tone correctly is $0 \leq \alpha \leq 1$, then, assuming independence of errors, the probability that in a sequence of N tones (n-gram) *all* tones are correct is α^N, which decreases rapidly. For example, with an accuracy of $\alpha = 0.8$ the probability to get a 10-gram correct is just $(0.8)^{10} = 0.11$. On the other hand, to achieve an accuracy of 0.8 for 10-grams the unigram accuracy should be at least $\alpha = 0.97$. This demonstrates that a MIR system which might be considered successful in regard to standard metrics might in fact be insufficient for evaluating secondary applications. Note-level errors can give rough estimates of n-gram errors, but only under idealized assumptions such as the independence of errors, which might not be fulfilled in real-world scenarios, as these depend, for instance, on audio quality, musical context, or instrument specifics. Furthermore, for research applications, rough estimates are not enough. For more complex evaluation measures, such as precision, recall, and F1 scores, this approximation might fail. Since we want to make heavy use of patterns, we need more precise numbers and propose an n-gram matching metrics where the estimated n-grams are matched and compared to ground-truth n-grams.

For a second metric, we use pattern commonalities which are a way to assess the common share of patterns between solos, say, the solos of two performers or two jazz styles. As such, it is a measure for influences between performers or jazz styles, and thus central for many of our research questions. Pattern commonality is based on the totality of all n-grams up to a certain maximal length N and is thus a statistical measure which might be tolerant to errors. This is a rather specific measure which might be only of marginal interest to others, but we need to know to what extent results based on AME transcriptions can be trusted.

Finally, we propose two more metrics, that are based on mimicking real-world pattern similarity search. Using similarity search could work as a counter-measure to transcription errors as the correct pattern might still be in the result set even if some errors occurred. Similarity search is also relevant for pattern mining in itself, as patterns in jazz might be performed with slight variations [11]. However, using similarity search can blow up the result set quickly, resulting in many uninteresting results, which might diminish the usability. This problem might become even more prevalent when using automatically derived estimates of the notes. The similarity scenario will be used to construct an evaluation metrics based on actual retrieval results and by estimating the extent of the "by-catch" in the search results depending on n-gram lengths and similarity thresholds. In this work, we employ two AME methods for applying the proposed evaluation metrics: convolutional-recurrent neural networks (CRNN) [3], which was specialized for and trained on monophonic jazz solos, and Melodia [23], a state-of-the-art but generic AME algorithm, which will serve as baseline. Furthermore, we introduce a novel Viterbi smoothing on top of the CRNN system in the hope of improving the performance in pattern search tasks.

2 AUTOMATIC DOMINANT MELODY ESTIMATION

Automatic melody estimation (AME) is defined as extracting the main melodic line from polyphonic music. Most proposed AME approaches are based on computing a time-frequency salience representation where F_0's of the main (dominant) melody line are emphasized [3, 5, 7, 23]. To obtain salient features for main melody F_0's against a polyphonic background is not trivial and is still considered the main bottleneck in melody estimation algorithms [3, 5].

To tackle this problem, several methods have been proposed, including hand-crafted features such as the harmonic sum spectrum (HSS) [23], representation learning methods such as source-filter non-negative matrix factorization (SF-NMF) [7, 9], and, more recently, deep learning based approaches [1, 3, 5, 24].

Due to the inherent temporal structure in music, a temporal tracking or smoothing phase is usually involved in AME systems [3, 6, 8, 23]. It is necessary to model the correlations between the frames. Popular choices for temporal tracking are HMMs [9, 22], contour tracking [7, 23], and recurrent neural networks [3, 19].

2.1 CRNN with SF-NMF pretraining

In [3], a convolutional-recurrent neural network was proposed whose pretraining is based on the SF-NMF model by [9]. In SF-NMF, the main melody is modeled by a source-filter model inspired by a model of speech production. The source is further decomposed into basis and activation matrices where the activations of the sources are considered as the initial pitch salience. This initial estimation is enhanced with a CNN and then a bi-directional RNN (Bi-GRU) is employed for modeling the temporal information. The classification layer outputs a frame-wise probability distribution over classes (target F_0's and a non-melody class). A simple peak-picking strategy is applied to the frame-wise probability distributions to obtain the F_0 estimates. The results show that when a good initial salience input is provided to the CRNN system, it performs considerably better without any augmentation or additional training data, hence, keeping a lower model complexity. The block diagram of the CRNN system is given in Fig. 1 (left).

2.2 Melodia

In Melodia [23], the salience representation is obtained by first detecting the sinusoidal peaks in the STFT and mapping each peak's energy to all harmonically related F_0 candidates in the STFT frame, with exponentially decaying weights. Then the salience

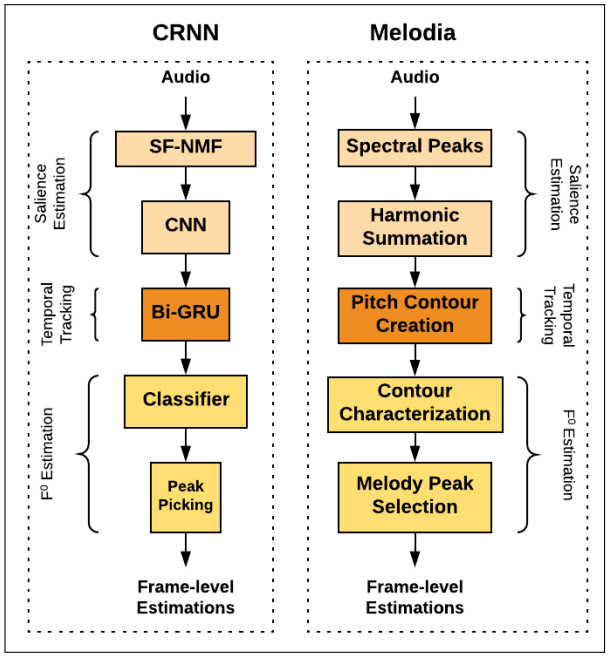

Figure 1: AME architectures: CRNN with SF-NMF pretraining (left) and Melodia (right).

peaks, which are continuous in time and frequency, are sequentially grouped to create contours which are treated as smooth melodic line candidates and are characterised by a set of features such as pitch and salience deviation. Finally, the melody contours are selected using a set of heuristic rules based on the contour characterizations. The block diagram of Melodia is given in Fig. 1 (right).

2.3 Viterbi smoothing with a time- and note-dependent transition matrix

As stated in Sec. 2.1, the CRNN system exploits the temporal information with an RNN layer (Bi-GRU) which is reported to increase the overall accuracy significantly. However, our initial experiments with this system show that the output F_0 estimations are not smooth enough, e. g., because of single frame estimation errors in a segment of frames. Such errors have minimal effect on the overall frame-level accuracy, but could negatively affect note-level and n-gram accuracy. This suggests that a simple peak-picking on the CRNN output probability distributions may not be sufficiently smooth for the pattern extraction task. As a possible improvement, we propose to replace peak picking with a novel *frame-level* Viterbi smoothing algorithm. Details can be found in Appendix A.

3 EVALUATION METRICS

We devise three evaluation metrics, where one is based on precision, recall and F1 scores, and the two other focus on performance in actual applications needed for musicological research on patterns in jazz, i. e., pattern search and pattern commonalities.

3.1 N-gram matching

This metrics is related to well-known note-based metrics in AME [21]. We work on a set of target solos (the ground truth in the form of acceptable transcriptions), that the AME algorithm is supposed to reproduce as closely as possible. The problem lies in the definition of closeness, which turns out to be a free parameter of the metric, and which is constrained by musicological criteria. This parameter δ is the width of a search window around the true onset of a note event. From music psychological research [10, 13], it is known that the just noticeable difference for onsets is in the order of 30–70 ms from which we can obtain the order of magnitude for errors in human annotation of onsets. This is also related to the speed of the fastest human movements (trills), which is about 16 Hz–20 Hz, implying an upper bound of about 50 ms–62.5 ms for inter-onset intervals. A grid search with $\delta \in \{30, 50, 70, 90\}$ showed that $\delta = 50\,ms$ gave the best results, so we used this value in the following. In many MIR evaluation metrics, thresholds of similar magnitude are employed, e. g. for beat tracking a threshold of 70 ms is customary [21].

To be more specific, let us define some basic notions. A melody μ of length N is defined as a sequence of onset t_i, pitch p_i, and duration d_i values, $\mu_i = (t_i, p_i, d_i)$ for $0 \le i < N$. Duration values are often of lesser importance, since for rhythm perception and production the inter-onset intervals are the functional time structure, whereas duration governs articulation. In pattern analysis of jazz, commonly only pitch or interval n-grams are considered. We will neglect duration here, and timing (onset) information needs to be "coarse-grained" in some way, which is achieved by using the $\delta = 50\,ms$ tolerance windows.

Given a ground truth reference solo in reduced representation, i. e., $\mu_i^T = (t_i^T, p_i^T)$ of length N, and an automatically extracted melody $\mu_i^Q = (t_i^Q, p_i^Q)$ of length M, we will now define recall, precision and F1 scores using a time window δ applying the following procedure.

For each tone event μ_k^T in the reference melody, we find all tone events in the estimated melody with onsets in a window of $\pm\delta$ around the onset t_k^T of the ground truth event, i. e., the set

$$M_k = \{Q_i \mid |t_i^Q - t_k^T| \le \delta\}. \tag{1}$$

If the set is empty, $M_k = \emptyset$, then we have a false negative and set $\mathrm{FN}_k = 1$, else $\mathrm{FN}_k = 0$. If the set contains exactly one element with the same pitch, we have a true positive TP_k. All other cases are counted as one false positive FP_k, even if there are more events, which in practice rarely happens due to the small time window and the smoothing step of the AME algorithms. This allows to get a unique triplet $(\mathrm{TP}_k, \mathrm{FP}_k, \mathrm{FN}_k)$ with $\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k = 1$ for each reference event. This procedure is not yet complete as it ignores extra events in the estimated melody outside of all time windows around the reference onsets. These events should be taken into account as extra false positives. To calculate these, we use the same procedure as before but with the roles of reference and estimated events interchanged while ignoring all but the false negative (unmatched) events which are added to the false positives in the final summation over a melody. Setting $\xi = \sum_k \xi_k$ for $\xi \in \{\mathrm{TP}, \mathrm{FP}, \mathrm{FN}\}$, precision, recall and F1 scores can then be defined in the usual way.

As we are not only interested in single tone events but also in patterns, the approach will be generalized to arbitrary n-grams. To

this end, we define an n-gram event for a melody as $\mu_i^n = (t_i^n, p_i^n)$ with

$$t_i^n = \frac{1}{n} \sum_{i \le k < i+n} t_k, \tag{2}$$

where t_i^n is the mean onset, and p_i^n

$$p_i^n = (p_i, p_{i+1}, \ldots, p_{i+n-1}), \tag{3}$$

the corresponding multi-pitch event (pitch n-gram). The choice of the mean onset for the onset of a multi-pitch event seems to be most principled, as it evens out small temporal variations of the onsets of the constituting pitches. Using the first onset of the n-gram would make the matching too strongly dependent on the error of a single event. Theoretically, this procedure allows n-grams with grossly mismatching onsets of the single pitches to match a ground truth n-gram, if only the mean onset matches. This is, however, not a problem, because it is very unlikely to happen and it would mean that preceding and succeeding n-grams will not match, which would have a strong negative impact on the metric.

From these definitions, we get a surrogate n-gram event sequence of $N - n + 1$ elements, where N is the length of the melody and n the length of the n-grams. The metrics from above can then simply be carried over for any value of n. This n-gram based approach has some similarity with the BLEU method [18] used for automatic translation, where also n-grams are matched but without regard to position.

3.2 Pattern search

In the "Dig That Lick" project, we developed two pattern search tools [12], the DTL Pattern Search[1], featuring regular expressions and two-staged searches, and the DTL Similarity Search[2], based on similarity of symbol sequences, derived from abstractions of the musical surface (e. g., pitches and intervals). The main idea of our second metrics is to compare the retrieval results for the ground truth with the retrieval results for the automatically extracted melodies using the similarity search.

For two symbol sequences $t = t_i^n$ and $q = q_j^m$ of lengths n and m, we define the edit distance similarity as

$$\sigma(t, q) = 1 - \frac{\eta(t, q)}{\max(n, m)}, \tag{4}$$

where $\eta(t, q)$ is the edit distance, the minimum number of substitutions, insertions, and deletions to transform one sequence into the other. Edit distances are a well-known family of sequence metrics which have been shown to provide good approximations for human similarity judgements, even using only single dimensions such as pitch, interval, and rhythm alone [14, 15], and are often used in pattern mining of melodies (e. g., [25, 26]). Hence, the good approximation property and the ease of implementation suits the needs of our pattern similarity search over a range of different melodic representations that can be chosen by the user. It should be noted that the evaluation procedure proposed below does not depend on the details of the employed similarity measure, it would work with any similarity measure.

The proposed metrics tries to mimic similarity searches over a range of query patterns with a range of similarity thresholds and maximum length differences as parameters. For each query and parameter setting, the search will retrieve all n-grams in the database that have a similarity to the query greater or equal to the specified threshold and with a length difference of at most the specified value. The result set also contains information on the containing solos for any retrieved instance. All searches are performed on the ground truth database as well as the corresponding database of estimated solos. For a similarity threshold of $\tau = 1$, the search is exact, which allows to define retrieval scores in the following way.

For each query, the result set is compared with the ground truth in terms of the set of solos in which the matches are found. For each solo in the result set from the ground truth database, containing the query n-gram, we compare the number of instances found in the ground truth (n_R) and in the corresponding solo in the estimated database (n_E). We set $n_E^i = 0$, if document i is not contained in the query result from the estimated database. We use the following partition of n_R as defining equation for true and false positives and false negative counts:

$$\begin{aligned} n_R &= TP - FP + FN \\ &= \min(n_E, n_R) - \theta(n_E - n_R) + \theta(n_R - n_E), \end{aligned}$$

with $\theta(x) = \max(x, 0)$ the ramp function. If $n_E \le n_R$ then we have $TP = n_E$, $FN = n_R - n_E$, and $FP = 0$; if $n_E > n_R$, then we have $TP = n_R$, $FP = n_E - n_R$, and $FN = 0$. These values are summed up across all solos in the ground truth result set, from which precision, recall and F1 scores can be calculated for a query.

Furthermore, we are interested in comparing the total sizes of result sets for a query in the ground truth and the AME databases, because for similarity search, results sets can grow quickly, and it is interesting to see if more or less results are generated for the AME databases. We use the relative difference of result set sizes with respect to the ground truth result set size as a metrics:

$$\Delta R(q_k; \tau, \delta) = \frac{\#R(q_k, D_R; \tau, \delta) - \#R(q_k, D_E; \tau, \delta)}{\#R(D_R; \tau, \delta l)}, \tag{5}$$

where $\#R(D_I; \tau, \delta)$ is the size of the result set from database I for the query q_k with similarity threshold τ and maximal length difference δ. Since these are relative differences, the values can be averaged across all queries to give overall evaluation values.

3.3 Pattern commonalities

Another application of interest in jazz pattern research is the assessment of similarity of subsets of solos with regard to pattern commonalities, e. g., comparing performer X with performer Y to investigate musical influence and stylistic closeness via shared patterns. We denote the set of all different n-gram values of length n for a set S as $\mathbf{n}(S)$. The pattern commonality of S and R for a fixed length n is then defined as the total variation distance of the relative frequency distributions:

$$\sigma_n(S, R) = \frac{1}{2} \sum_{q \in \mathbf{n}(S \cup R)} |f_S(q) - f_R(q)|, \tag{6}$$

where $f_X(q)$ is the relative frequency of n-gram q in $X = S, R$, and we set $f_X(q) = 0$ for $q \notin \mathbf{n}(X)$. This can be interpreted as the half of the L_1-norm of the difference vector of n-gram frequencies over the joint n-gram set. There are many different possibilities to measure commonalities of two sets, e. g., the Jaccard similarity and

[1] https://dig-that-lick.hfm-weimar.de/pattern_search/
[2] https://dig-that-lick.hfm-weimar.de/similarity_search/

its many variants. However, informal experiments showed that all these options are strongly correlated with each other, so we decided to use total variation distance as it has a better numerical solution and is somewhat easier to compute.

To compare the ground truth and the extracted melodies, one can calculate the pattern commonalities over a range of subsets (solos) and n-gram lengths, and use the mean of absolute relative difference (compared to the ground truth) as a metrics for how well the estimated values approximate the true pattern commonalities. Formally, for two corresponding sets S_R, S_E with K elements μ_i, μ'_i,

$$\sigma_n(S_R, S_E) = \frac{2}{K(K-1)} \sum_{i<j} \frac{|\sigma_n(\mu_i, \mu_j) - \sigma_n(\mu'_i, \mu'_j)|}{\sigma_n(\mu_i, \mu_j)} \tag{7}$$

In practice, one will not compute all pairings but use a random sample. Moreover, for large n, the frequency distributions are more and more degenerated, with very few common n-grams in any pair of solos. Hence, it will be reasonable to restrict the evaluation to lower values of n. Additionally, true and estimated pattern commonalities can be correlated to yield another metrics.

4 RESULTS

In this section, we demonstrate the proposed evaluation metrics by comparing the melody estimation performances of CRNN, CRNN with Viterbi smoothing (CRNN-V) and Melodia algorithms on the jazz solo recordings from the Weimar Jazz Database [20]. The WJD contains 456 transcribed and annotated solo recordings (~13 hours) from 78 artists with 13 different types of instruments, spanning a time range from 1925 to 2009. The pitches are transcribed with MIDI numbers, hence, we use a semitone resolution for both methods.

4.1 Training and Initialization

We trained the CRNN from scratch using only jazz solo tracks from the WJD in order to have a system specialized for jazz music. For this reason, we created train, validation, and test sets following an artist-conditional random partitioning as described in [7]. This resulted in 290, 75, and 91 tracks in train, validation, and test sets, respectively. All tracks were initially downsampled to 22050 Hz sampling rate and the time resolution was set to 11.6 ms for all algorithms. The configuration of CRNN as well as the parameters of SF-NMF were chosen following [3] where the target F_0 range is between A1 = 55 Hz and A6 = 1760 Hz. For Melodia, we tried different sets of parameters for optimizing the accuracy, but the highest accuracy was obtained with the default parameters.

For the Viterbi smoothing algorithm, we calculated the frame-level and note-level state transition probabilities from the train and validation sets by simply counting the occurrences of each transition with subsequent normalization. To convert the frame-level pitch track into a note track, we first converted F_0 from Hertz to MIDI values and then formed notes by simply grouping consecutive frames with the same MIDI values. For all evaluations, we used the test set with 51,825 note events in 91 solos from 14 artists playing four different instruments, covering a time range from 1938 to 2009. We calculated all pitch n-grams up to length 10, which resulted in 511,725 n-grams in total.

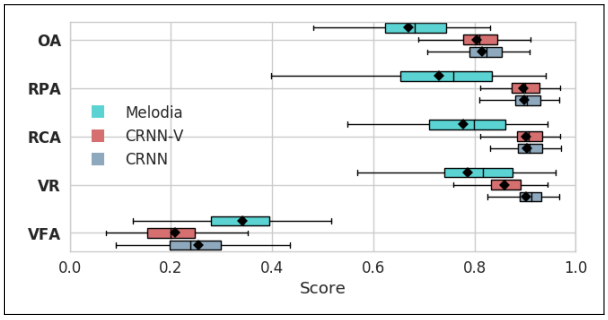

Figure 2: Frame-wise evaluations of CRNN, CRNN-V and Melodia.

4.2 Frame-wise performance analysis

AME systems are usually evaluated with a frame-wise comparison to the ground-truth using overall accuracy (OA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), voicing recall (VR), and voicing false alarm rate (VFA) metrics as defined in [21]. Here, we compare the performances of CRNN, CRNN-V and Melodia systems on a test set with 91 solos from the Weimar Jazz Database [20]. The evaluation results, given in Figure 2, show that, unsurprisingly, the CRNN and CRNN-V are significantly better than Melodia for all metrics. This is due to training CRNN directly on jazz tracks whilst Melodia performs a more generic melody estimation. CRNN and CRNN-V have similar performances on RPA and RCA. CRNN has a better OA and VR but worse VFA (higher). This is actually not unexpected due to the nature of Viterbi smoothing. As the aim is to obtain more precise estimations for the pattern mining task, Viterbi smoothing results in a much higher rate of non-melody frames than simple peak picking, that eventually decreases VFA but also VR. It can be concluded that CRNN performs slightly better than CRNN-V, however, this does not indicate how well these systems perform for pattern extraction.

4.3 N-gram-based evaluations

4.3.1 N-gram matching evaluation. N-gram matching evaluation was carried out according to the algorithm in Sec. 3.1 for each AME system. The resulting precision, recall, and F1 scores are depicted in Fig. 3. The CRNN models attain very high values for small N (cf. Tab. 1) and keep a rather good performance for large N. Melodia on the other hand has $F_1 = .59$ for unigrams and effectively no correct recall at $N = 10$. The curves also show that for CRNN systems the decrease in performance with increasing N is much slower than geometrical, whereas Melodia comes quite close to this behaviour. This indicates that the errors for the CRNN systems are not independent from each other, which is further corroborated by the correlation coefficients of the n-gram matching and the frame-wise overall accuracy, which rapidly decreases for larger N. As stated earlier, the frame-wise evaluation can be used as an approximation for note-wise evaluation scores, but it works better for recall and F1 scores than for precision. However, from Tab. 1, it seems that these correlations are themselves correlated with the n-gram-based scores, so frame-wise accuracy might be only a good approximation if overall accuracy is already very high. This justifies

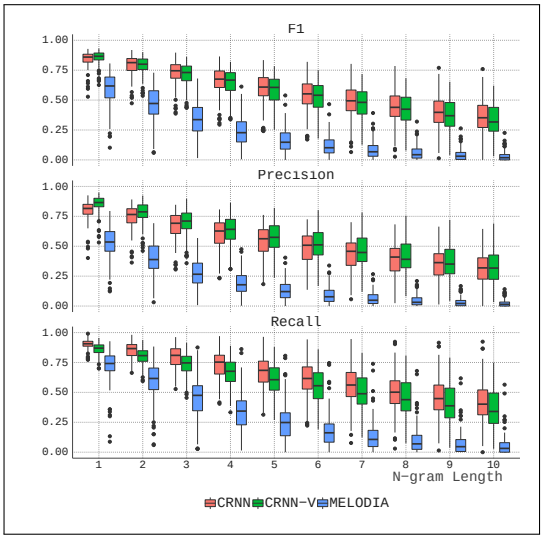

Figure 3: F1, precision, and recall values for the n-gram matching metrics for CRNN, CRNN-V and Melodia.

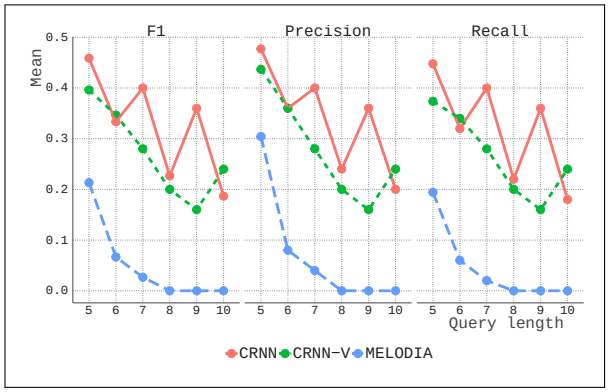

Figure 4: CRNN, CRNN-V and Melodia retrieval results (F1, precision, and recall) for exact pattern search ($\tau = 1$) and a set of 25 search queries for each length value.

that our approach to generalize the matching evaluation to n-grams of arbitrary length is necessary for good estimation of pattern retrieval performances based on AME systems. One thing to note is that precision for the CRNN-V system is systematically higher than for the non-Viterbi version, which shows that the optimization for precision was successful, even though this happens at the expense of slightly lower recall and F1 scores.

4.3.2 Similarity search retrieval values. For the similarity search evaluation, we used a randomly selected set of 150 queries with n-gram lengths in the range of 5 to 10 in order to simulate realistic application scenarios where one rarely searches for very short patterns. For each length condition, we randomly picked 25 queries from the ground truth test set without further constraints. For each query, we ran a similarity search for each possible combination of similarity thresholds $\tau \in \{0.5, 0.6, \ldots, 1\}$ and three maximum

N	AME	F1	r_{F1}	prec	r_{prec}	rec	r_{rec}
1	CRNN	0.84	0.73	0.79	0.69	0.90	0.66
1	CRNN-V	0.85	0.68	0.85	0.65	0.86	0.60
1	MEL	0.59	0.40	0.52	0.36	0.71	0.39
5	CRNN	0.59	0.64	0.54	0.67	0.66	0.51
5	CRNN-V	0.58	0.59	0.57	0.67	0.60	0.41
5	MEL	0.17	0.44	0.13	0.40	0.25	0.39
10	CRNN	0.35	0.58	0.31	0.63	0.41	0.46
10	CRNN-V	0.34	0.55	0.33	0.64	0.36	0.37
10	MEL	0.03	0.40	0.02	0.38	0.06	0.36

Table 1: F1, precision (prec), and recall (rec) scores from the n-gram matching evaluation along with correlation coefficients with overall frame-wise accuracy (OA) for a selection of n-gram lengths N.

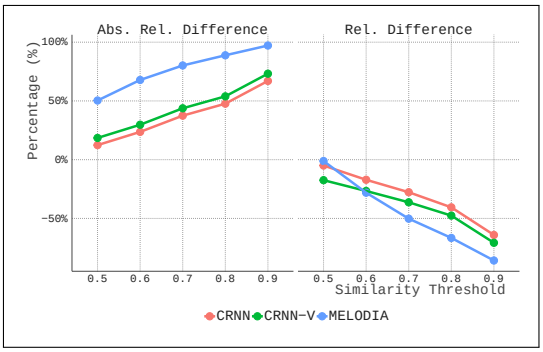

Figure 5: Relative result set sizes for similarity search with $\tau < 1$. All values averaged across queries of length 7 and max. length differences. Left: Averaged absolute relative changes in result sizes of AME compared to the ground truth. Right: Averaged relative changes.

length differences $\delta \in \{0, 1, 2\}$. Separate searches were run over the n-gram databases derived from the ground truth and the AME systems, and evaluation scores given in Sec. 3.2 were calculated.

The similarity search retrieval values for the exact search with similarity thresholds $\tau = 1$ (implying also a maximum length difference of 0) for the three different systems can be seen in Fig. 4. As expected, the CRNNs outperform the Melodia estimation by a large margin. Retrieval scores for the CRNN systems start with values between 0.4 and 0.5, as could be expected from the n-gram matching evaluation scores, and decrease rather steeply with query length. The zigzag pattern for the CRNN version is rather puzzling, and also the increase from query lengths 9 to 10. But this is probably due to the specific set of queries or a result of the Viterbi smoothing.

The relative difference of result set sizes for all other cases with $\tau < 1$ is depicted in Fig. 5 for queries of length 7, graphs for the other lengths are qualitatively similar. The relative differences of result set sizes increase with the similarity threshold, being on average about 55 %–60 % smaller for $\tau = .90$, but nearly of the same size for $\tau = .50$. Since the averaged absolute change is larger than the averaged signed difference, this is pattern dependent. However, on average *less* by-catch is retrieved by the AME systems. This is

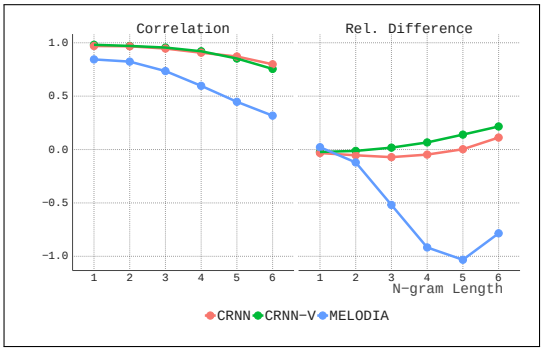

Figure 6: Results of comparing pattern commonalities for n-gram lengths 1–6. Left: Spearman correlation coefficients between estimated and ground-truth commonalities. Right: Relative differences of estimated and ground-truth values with regard to ground-truth values. Positive/negative values mean that pattern commonalities are over/underestimated.

probably due to the fact that the transcription errors introduced by the AME systems are of a different nature than the pattern variations used by the performers.

4.3.3 Pattern commonalities evaluation. For 10 batches with 10 randomly selected performers each, we calculated pattern commonalities between all possible pairs for n-gram lengths 1 to 6. This was done separately for the ground truth and the three AME systems. From this, we calculated Spearman correlations between similarity values obtained from the ground truth and the AME estimations. Furthermore, we calculated relative differences of the AME values with respect to the ground truth values. Results can be seen in Fig. 6. For the CRNN models correlations are very high and stay rather high with increasing n-gram length ($r_{N=6}$ = .80 for CRNN and $r_{N=6}$ = .76 for CRNN-V). For Melodia, correlations are also rather high for small N but drop quickly ($r_{N=6}$ = .32). Consequently, the mean relative differences are close to zero for the CRNN systems and short n-grams, but for larger N they are overestimating pattern commonalities, whereas Melodia grossly underestimates the pattern similarity values already for N = 3.

5 DISCUSSION

In this paper, we proposed several new evaluation methods for AME systems driven by the needs of musicological applications in jazz research. We demonstrated that a specialization of AME systems for specific data sets and purposes is an inevitable step for high-quality research applications. General algorithms trained on publicly and readily available datasets from a different musical domain are unlikely to work out of the box for a genre like jazz. We also showed that evaluations should be tailored for secondary tasks and derived features beyond the primary goal of estimating the main melody. Whereas the retrieval values for note tracks are rather satisfying for small to moderate n-gram lengths, it is very hard to retain this good measure for longer n-gram lengths, which is, of course, a function of quality of the primary note estimation, which has to be very high *ab initio*. Furthermore, frame-wise evaluation metrics only provide a good estimate for note-based evaluation, if the quality of extraction is high, but not for n-gram-based measures. On the other hand, for certain derived tasks of a more statistical nature which average across many events, such as pattern commonalities, even sub-par performances of AME systems might still give reasonable and usable results, but again only, if a certain level of quality is reached. The Melodia note estimations, for example, are not sufficient in this application scenario at all.

For pattern research, even a single frame error can be destructive depending on when it occurs, e. g., inside a note event. As a result, the precision of an AME system according to the n-gram matching metrics is one of the most relevant indicators about how it will behave on pattern search tasks. The CRNN system already has a temporal modelling via RNN stage but the results show that CRNN-V has a higher precision for all n-grams. Hence, the proposed Viterbi smoothing, even not better than the CRNN for F1 and recall, might be useful if precision is the main goal.

Even though the CRNN systems show a highly improved performance, there is still room for further improvements. As an example, current AME systems are devised to find the 'dominant' melody. However, in jazz solo tracks, the dominant melody may not be necessarily the soloist all the time, i. e., accompanying instruments such as piano and bass can become dominant for small time intervals. Such notes from the background, although being dominant, have to be correctly estimated as non-melody from a pattern search point of view. One solution for this problem is to integrate instrument information into AME systems to be able to extract only notes from particular instruments. Especially for jazz tracks, meta-data is available through jazz discographies such as Tom Lord [3] or Linked Jazz [4] and solo instrument information can also be retrieved from such sources. First experiments showed that using (frame-based) instrument information was able to boost note-wise accuracy beyond the 90 % sonic wall. Furthermore, all evaluation results reported here are average values, but there is in fact considerable variation in the AME results, as can be seen in the boxplots in Fig 3. It would be desirable to be able to predict the AME quality for a solo from available metadata or from audio features. We assume that recording quality (e. g., the prominence of the solo instrument in the mix) as well as instrument timbre qualities of certain performers (e. g., strong vibrato) have strong influences on AME performance.

ACKNOWLEDGMENTS

The "Dig That Lick" project was funded by the "Diggin' into Data Challenge" as DFG project PF 669/9-1 (Germany), ESRC project ES/R004005/1 (UK), ANR project ANR-16-DATA-0005 (France), and NEH project NEH-HJ-253587-17 (USA).

REFERENCES

[1] Jakob Abeßer, Stefan Balke, Klaus Frieler, Martin Pfleiderer, and Meinard Müller. 2017. Deep Learning for Jazz Walking Bass Transcription. Erlangen, Germany.
[2] Stefan Balke, Jonathan Driedger, Jakob Abeßer, Christian Dittmar, and Meinard Müller. 2016. Towards Evaluating Multiple Predominant Melody Annotations in Jazz Recordings.. In *ISMIR*. 246–252.
[3] D. Basaran, S. Essid, and G. Peeters. 2018. Main Melody Estimation with Source-Filter NMF and CRNN. In *19th International Society for Music Information Retrieval Conference (ISMIR)*.
[4] Paul F. Berliner. 1994. *Thinking in Jazz. The Infinite Art of Improvisation.* University of Chicago Press, Chicago.

[3] https://www.lordisco.com/
[4] https://linkedjazz.org/

[5] R.M. Bittner, B. McFee, J. Salamon, P. Li, and J.P. Bello. 2017. Deep salience representations for F_0 estimation in polyphonic music. In *18th International Society for Music Information Retrieval Conference (ISMIR)*.
[6] R.M. Bittner, J. Salamon, S. Essid, and J.P. Bello. 2015. Melody extraction by contour classifi- cation. In *15th International Society for Music Information Retrieval Conference (ISMIR)*.
[7] J.J. Bosch, R.M. Bittner, J. Salamon, and E. Gómez. 2016. A Comparison of Melody Extraction Methods Based on Source-Filter Modelling. In *17th International Society for Music Information Retrieval Conference (ISMIR)*.
[8] J. L. Durrieu, B. David, and G. Richard. 2011. A Musically Motivated Mid-Level Representation for Pitch Estimation and Musical Audio Source Separation. *IEEE Journal of Selected Topics in Signal Processing* 5, 6 (Oct. 2011), 1180–1191. https://doi.org/10.1109/JSTSP.2011.2158801
[9] J. L. Durrieu, G. Richard, B. David, and C. Fevotte. 2010. Source/Filter Model for Unsupervised Main Melody Extraction From Polyphonic Audio Signals. *IEEE Transactions on Audio, Speech, and Language Processing* 18, 3 (March 2010), 564–575. https://doi.org/10.1109/TASL.2010.2041114
[10] Paul Fraisse. 1963. *The psychology of time.* Harper & Row, New York.
[11] Klaus Frieler, Frank Höger, and Martin Pfleiderer. 2019. Anatomy of a Lick: Structure & Variants, History & Transmission. In *Book of Abstracts: Digital Humanities 2019, Utrecht*. Utrecht, Netherlands. https://dev.clariah.nl/files/dh2019/boa/0638.html
[12] Klaus Frieler, Frank Höger, Martin Pfleiderer, and Simon Dixon. 2018. Two web applications for exploring melodic patterns in jazz solos. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR), Paris 2018*. Paris.
[13] Brian C. J. Moore. 1995. *Hearing*. Academic Press, San Diego.
[14] Daniel Müllensiefen and Klaus Frieler. 2004. Cognitive Adequacy in the Measurement of Melodic Similarity: Algorithmic vs. Human Judgments. *Computing in Musicology* 13 (2004), 147–176.
[15] Daniel Müllensiefen and Klaus Frieler. 2007. Modelling Experts' Notion of Melodic Similarity. *Musicae Scientiae: Similarity Perception in Listening to Music* Discussion Forum 4A (2007), 183–210.
[16] Martin Norgaard. 2014. How Jazz Musicians Improvise: The Central Role of Auditory and Motor Patterns. *Music Perception: An Interdisciplinary Journal* 31, 3 (Feb. 2014), 271–287. https://doi.org/10.1525/mp.2014.31.3.271
[17] Thomas Owens. 1974. *Charlie Parker. Techniques of Improvisation.* PhD Thesis. University of California, Los Angeles.
[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, PA, USA, 311–318.
[19] H. Park and C. D. Yoo. 2017. Melody extraction and detection through LSTM-RNN with harmonic sum loss. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2766–2770. https://doi.org/10.1109/ICASSP.2017.7952660
[20] Martin Pfleiderer, Klaus Frieler, Jakob Abeßer, Wolf-Georg Zaddach, and Benjamin Burkhart (Eds.). 2017. *Inside the Jazzomat - New Perspectives for Jazz Research.* Schott Campus, Mainz.
[21] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. 2014. mir_eval: A transparent implementation of common MIR metrics. In *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer.
[22] M. P. Ryynanen and A. Klapuri. 2005. Polyphonic music transcription using note event modeling. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.* 319–322. https://doi.org/10.1109/ASPAA.2005.1540233
[23] J. Salamon and E. Gomez. 2012. Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 6 (Aug. 2012), 1759–1770. https://doi.org/10.1109/TASL.2012.2188515
[24] Li Su. 2018. Vocal melody extraction using patch-based CNN. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 371–375.
[25] Pieter v van Kranenburg. 2010. *A computational approach to content-based retrieval of folk song melodies.* PhD Thesis. Unisversity of Utrecht, Utrecht, Netherlands.
[26] Anja Volk and Peter van Kranenburg. 2012. Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae* 16, 3 (Nov. 2012), 317–339. https://doi.org/10.1177/1029864912448329

A VITERBI SMOOTHING WITH A TIME- AND NOTE-DEPENDENT TRANSITION MATRIX

In a simple Viterbi algorithm, the current frame-level state at frame t is conditioned only on the previous frame-level state at frame $t-1$. Our novel contribution here is that we model the note-level state transitions as well and condition the current frame-level state also on the previous note-level state. The rationale behind this comes from the inherent structure in music, where the consecutive notes in a piece are correlated. One can utilize this information in computing the next frame-level state. In addition, we account for the phenomenon that the dependency between notes decreases with increasing time difference (offset-onset interval) in modelling note-level state transitions.

Formally, we denote output note classes of the CRNN by $\mathbf{U_m} = \{u_{1:U}\}$, the non-melody class by u_0 and the list of all classes by $\mathbf{U}$. We also denote the frame-level hidden state at frame t by $s^f_t \in \mathbf{U} = \{u_{0:U}\}$ and the note-level state by $s^n_{t_1,t_2} \in \mathbf{U_m}$ where t_1 and t_2 represent the onset and offset frames respectively. We define the note-level state transition probability as

$$p(s^n_{t_3,t_4}|s^n_{t_1,t_2}, t_3 - t_2) \quad \text{where} \quad t_2 < t_3. \tag{8}$$

Then, we define the time- and note-dependent transition probability distributions as

$$p(s^f_t|s^f_{t-1}, s^n_{t_1,t_2}, t - t_2) \quad \text{where} \quad t_2 < t. \tag{9}$$

Note that the dependency on the previous note state $s^n_{t_1,t_2}$ and time difference drops for $s^f_t = u_0$, since (8) does not model transitions to the non-melody state. On the other hand, for $s^f_t \neq u_0$, we choose the expressions in (9) and (8) to be equal at $t = t_3$, so that $s^n_{t_3,t_4} = s^f_t = u_i$, where $u_i \in \mathbf{U_m}$.

Both note-level state transitions and frame-level state transitions can be computed from the available data by simply counting the occurrences. Analysis of the frame-level transitions in the Weimar Jazz Database (WJD) [20] shows that a direct transition from a note to another note happens in less than 0.05% of the transitions. As a result, we may further simplify the model by keeping the dependency on the previous note-level state only for the case where $s^f_t \neq u_0$ and $s^f_{t-1} = u_0$.