Gabriel Durán
and Patricio de la Cuadra
Departamento de Ingeniería Eléctrica
Pontificia Universidad Católica de Chile
Vicuña Mackenna 4860, Santiago 8940000
Chile
{geduran, pcuadra}@uc.cl

Transcribing Lead Sheet-Like Chord Progressions of Jazz Recordings

Abstract: The vast majority of research on automatic chord transcription has been developed and tested on databases mainly focused on genres like pop and rock. Jazz is strongly based on improvisation, however, and the way harmony is interpreted is different from many other genres, causing state-of-the-art chord transcription systems to achieve poor performance.

This article presents a computational system that transcribes chords from jazz recordings, addressing the specific challenges they present and considering their inherent musical aspects. Taking the raw audio and minor manually obtained inputs from the user, the system can jointly transcribe chords and detect the beat of a recording, allowing a lead sheet–like rendering as output.

The analysis is implemented in two parts. First, all segments with a repeating chord progression (the chorus) are aligned based on their musical content using dynamic time warping. Second, the aligned segments are mixed and a convolutional recurrent neural network is used to simultaneously detect beats and transcribe chords.

This automatic chord transcription system is trained and tested on jazz recordings only, and achieves better performance than other systems trained on larger databases that are not jazz specific. Additionally, it combines the beat-detection and chord transcription tasks, allowing the creation of a lead sheet–like representation that is easy to interpret by both researchers and musicians.

Introduction

Pitch-related tasks are not trivial, because pitch is closely related to perception and psychoacoustics. Moreover, studies on musical harmony, which can be understood as a subset of pitch-related tasks, are complex, as there is some amount of ambiguity involved. One example is automatic chord transcription (ACT), in which ground-truth transcriptions performed by different trained musicians might not be consistent (Ni et al. 2013; Humphrey and Bello 2015), because there is a process of subjective interpretation involved. Nevertheless, this challenging task has been widely addressed in the music information retrieval community and many ACT systems have been proposed in the last two decades (Pauwels et al. 2019).

Variations of chord-sequence prediction systems can be found in the literature with different names: recognition, identification, detection, transcription, or estimation. Nevertheless, the term transcription refers to a more-complex task closely related to musical high-level concepts and to functional analysis, as stated by Humphrey and Bello (2015). From that perspective, ACT systems should not analyze isolated time segments as independent, because the transcription of a particular chord depends strongly not only on its own musical content, but also on the observation of previous and subsequent chords.

Most research up to now has been developed on datasets like Isophonics and Billboard, which are highly biased towards genres like pop and rock. Jazz presents elements inherent in its nature that are mostly based on improvisation, causing some common assumptions to cease being valid regarding the way chords are manifested and how harmony is expressed. For example, chords are not played necessarily as they are written nor change exactly when they are expected to. The effects and limitations of ACT on jazz music has only been addressed, to our knowledge, by Eremenko et al. (2018), who presented and analyzed a large jazz database. In that work, they tested two state-of-the-art ACT systems that are not specifically trained for jazz music, showing drastically lower performance than on pop-rock databases—around half of the accuracy—demonstrating that those approaches are not well suited for this genre.

Computer Music Journal, 44:4, pp. 26–42, Winter 2020
doi:10.1162/COMJ_a_00579

The article is structured as follows: First we introduce the musical context of this research, provide an in-depth description of our proposed method to ACT, and then present a literature review. Second, the segment-alignment task is presented and tackled. Third, the chord transcription implementation is described. Finally, experiments and results are presented.

Musical Context

Unlike many other genres, jazz is strongly based on improvisation. Spontaneity is common on the extensive instrumental solos, and each repetition of a segment is somehow different from the others. The way musicians play is highly related to what others are playing, generating a spontaneous performance that cannot be repeated identically.

Traditionally, the structure of jazz performances consists of a chord progression that acts as the "backbone" of the recording and has a fixed number of bars: Known as a *chorus*, it is used as the base on which musicians improvise during solos. It is not a formally established term, but is colloquially used by jazz musicians and it is worth noting that in other musical contexts the term chorus has different meanings and should not be confused. In jazz, the way chords are played is mostly improvised and the chorus is repeated multiple times, but each iteration is different: The chords played in one repetition of the chorus are unlikely to be exactly the same as the chords in another repetition, but all of them express some elements of the global harmonic progression. Normally, the theme, or main melody of the song, is presented in the first instance of the chorus, followed by instrumental solos improvised on the chorus's chord progression, and the last repetition consists of a restatement of the theme.

The chorus does not define the chords strictly, because they are rarely played as written (Pachet, Suzda, and Martínez 2013), instead it defines the chords as a basic form. Thus, the written chord progression is used as a reference and each musician will ornament the chords at each repetition—for example, adding or subtracting notes to a chord, or by anticipating or delaying it in time. There are some well-known techniques for substituting a chord in place of another functionally equivalent chord, or using a different chord that maintains some of the characteristic notes from the original (for example, tritone substitution, which maintains the third and the seventh of a dominant chord while transposing the root by a tritone). In jazz the chorus is usually represented in a *lead sheet* that contains the "essence" of a tune (Pachet, Suzda, and Martínez 2013), namely, the melody in music notation, along with chord names above the musical staff.

In jazz, the line of the double bass or bass guitar is often found as a *walking bass*, a technique in which notes of equal duration are steadily played, usually quarter notes (sometimes half or eighth notes are used as well). The pitches played are mostly part of the chord (root, third, fifth, seventh, etc.), but other pitches can be added, such as diatonic scale tones or chromatic alterations. Some jazz subgenres do not use the walking bass technique, but it can be found in the majority of "straight-ahead" (or mainstream) jazz recordings, on which this research is focused.

Proposed Method

We start from the idea that chorus repetitions should not be considered as independent musical segments, instead they all need to be taken into consideration to retrieve the most important harmonic elements presented in each one. This approach can be understood as a noncausal global analysis (so not real time), ignoring the details in the harmonic and melodic content not shared between choruses.

This method needs all the time frames in which each chorus begins and ends. This can be automated by implementing a complex musical structure-analysis task that is far from trivial. To focus on chord transcription, the structural analysis is set aside and the chorus boundaries are considered to be an input to this system.

The first step is to align all choruses in a recording based on their musical content. Choruses in jazz performances usually have slightly different lengths, because the tempo does not stay exactly constant, but all of them have the same number of bars. Thus, they need to be aligned prior to chord transcription,

matching their temporal frames and later warped in time so they have equal length.

The second step consists in chord transcription, in which the musical content of all choruses is analyzed at once by a deep-learning model, producing a single chord sequence. A beat detector is included in the same model as a subtask, making it possible to match chord transitions with beats and bars. This allows the predicted chord progression to be rendered as a lead sheet–like representation, making it easier to visualize and more familiar to musicians.

Literature Review

Automatic chord transcription is a highly active research topic that has been addressed since the first system was proposed by Takuya Fujishima (1999). The topic is nontrivial and, although state-of-the-art systems have achieved better results since deep-learning approaches were introduced (Pauwels et al. 2019; McFee and Bello 2017), there are many issues that are not yet resolved, such as handling subjective interpretation of chords, dealing with highly imbalanced databases, and ensuring consistency between chords in long harmonic progressions. Most traditional approaches comprised two stages: First, musically meaningful features are extracted from the audio, and second, a sequence analysis is performed and each temporal frame is assigned to a single chord-class from a predefined chord dictionary. The Chromagram, and its variations, are the most used features for the first stage (Bello and Pickens 2005; Sumi et al. 2008; Mauch 2010; Ni et al. 2012), as they describe the harmonic content on a low-dimensional vector. Hidden Markov models were first used for harmonic sequence analysis by Sheh and Ellis (2003) and became popular for ACT (Ni et al. 2012; McVicar et al. 2014), combined with Viterbi decoding.

Deep-learning methods have been used for the feature-extraction step, for sequence analysis, and more recently for both at the same time. In the work presented by Humphrey and Bello (2012), convolutional neural networks (CNNs) were used to learn features directly from a constant-Q transform (CQT) representation (Brown 1991), showing that simple network architectures achieve state-of-the-art results without performing posterior sequential analysis. Later, Boulanger-Lewandowski, Bengio, and Vincent (2013) introduced recurrent neural networks (RNNs) for sequence analysis, modeling temporal continuity, harmonic relations, and temporal dynamics. Recent work has successfully integrated both stages into a single system that is capable of learning musically meaningful features from a spectrogram-like representation and modeling temporal relations between frames. Generally, they combine CNN and RNN (McFee and Bello 2017; Jiang et al. 2019; Wu, Carsault, and Yoshii 2019), although the work presented by Korzeniowski and Widmer (2016) implemented conditional random fields for sequence decoding.

Past research on music synchronization frequently focused on the features used to retrieve meaningful information from the audio (Bello and Pickens 2005; Ewert, Müller, and Grosche 2009; Müller, Ewert, and Kreuzer 2009; Izmirli and Dannenberg 2010). This work is all based on the Chromagram as a feature (which represents the pitch content of a temporal segment in terms of the intensity of each of the twelve pitch classes of the chromatic scale), but presents some variations. For example, Müller and colleagues make pitch classes more invariant to changes in timbre by computing the mel-frequency cepstral coefficients (MFCCs) and removing their lower components prior to obtaining the Chromagram. Ewert and coworkers enhanced the Chromagram with note onsets, obtaining a Chroma-onset representation that led to better synchronization for music with clear note attacks. Wang et al. (2014) presented a novel method to align multiple sequences, simultaneously leading to more-robust results using Chroma-onset features. All these approaches rely on a version of the dynamic time-warping algorithm (DTW) to perform the alignment, but Maezawa et al. (2014) presented a probabilistic generative framework to align multiple performances of a recording. Nevertheless, these previous approaches implicitly assume that the audio recordings that will be aligned contain the same melody and chords but differ in their instrumentation, tempo, and recording conditions.

Focusing on semi-improvised music, Duan and Pardo (2011) presented a system to align audio with

its lead sheet. Because these recordings include improvisations that are not transcribed, the recordings are not fully annotated as MIDI scores. The authors propose an algorithm that aligns different audio versions to their lead sheet, based on beat-synchronous Chromagrams.

Contribution

This work introduces a novel approach to chord transcription in jazz music, taking into consideration some of its distinctive musical attributes, like improvised solos and a repeating harmonic progression. First, we perform a study on the musical features used for chorus alignment, to find those most suitable for this particular task. Second, a convolutional recurrent neural network (CRNN) architecture is used to jointly perform chord transcription and beat detection, while exploring three approaches to pool all the aligned choruses to a single one.

From a practical point of view, the proposed system is an aid to amateur musicians and jazz enthusiasts to retrieve the chord progression from a jazz performance as a lead sheet–like representation, which is easy to interpret. Even experienced musicians may benefit from this system, allowing them to automatize the chord transcription process and spend less time manually transcribing chords.

Chorus Alignment

The main goal of music alignment systems (the term music synchronization is often used interchangeably) is to find the optimal temporal alignment between two musical segments. This process compares each frame in one segment to each frame in the other, calculating their similarities, and relies strongly on the resemblance between frames in a given feature space. It is not usual to perform musical alignment over the raw waveform, this would be impractical owing to the high number of samples and lack of robustness against changes in timbre, intensity, etc. In general, the raw audio is transformed to a feature space containing higher-level content that unveils similarities between the audio signals. A popular algorithm used for audio alignment is the DTW (Ewert, Müller, and Grosche 2009), which finds the optimal path efficiently using dynamic programming.

Choruses in a jazz performance have the same harmony but do not share melody or bass line, and thus the similarities take place on a higher musical level, making alignment a more difficult task. The Chromagram, or its variations, are not necessarily the most suitable musical feature and new alternatives for alignment of jazz choruses should be investigated.

In this section, we present a study of musical features used for jazz chorus alignment, including melodic and harmonic features as well as timbre descriptors. The key idea is that a single feature type does not extract enough high-level information to describe such sophisticated musical data, and a feature set of descriptors that unveil different aspects of the music should be used. We also present a concise review of distance metrics used to compare the similarity between data.

Proposed Musical Features

The DTW algorithm will work properly only if two sequences are somehow similar in their feature space, so the choice of adequate features is crucial. The Chromagram (in the following referred to simply as "Chroma") captures melodic and harmonic information and is usually used for music alignment by virtue of its simplicity, low dimensionality, and ease of interpretation as musical pitches are represented explicitly.

One of the main disadvantages of Chroma features is that musical relations between notes are not considered, meaning that the similarity between one note and all others is equal. The tonal centroid features proposed by Harte, Sandler, and Gasser (2006), sometimes called Tonnetz features, are a nonlinear transformation of the Chromagram that reduces it to a 6-D feature space modeling relations between notes. For example, the Euclidean distance between notes that are a fifth or a third apart is small, and that between notes a tritone apart is larger, as shown in Figure 1.

Figure 1. Euclidean distance of each note compared to C in the Tonnetz feature space.

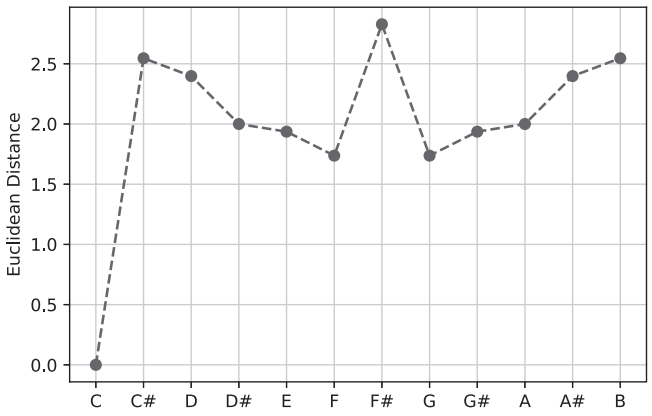


Table 1. Frequency Ranges Used for Feature Extraction

	Limits	
Frequency Range	*Hz*	*Pitch*
Low	[32 .. 261]	C1–C4
Mid	[130 .. 1046]	C3–C6
Treble	[523 .. 4186]	C5–C8

Both Chroma and Tonnetz features discard the frequency range from which each pitch came, making all octaves equally relevant. This is not consistent with the way chords and melodies are usually played—for example, it is unusual to play chords on a high pitch range. As different octaves contain different musical content, we propose splitting the frequency range into three overlapping bands and computing a Chroma and Tonnetz for each band. The frequency limits of each range are shown in Table 1 and they span almost the whole piano range (from C1 to C8). The bass frequency range is similar to the one used by Ryynänen and Klapuri (2007) and mostly matches the pitch range of a double bass or bass guitar. The other two frequency ranges were selected to have three octaves as well, and adjacent ranges overlap one octave, adding some redundancy.

Timbre is a complex musical quality related to human perception, and is independent of harmony and melody. Nevertheless, similarities between some specific musical events having similar timbre can be revealed through timbral features. The common descriptors used to model timbre are the lower coefficients of the MFCCs (Müller, Ewert, and Kreuzer 2009), so the first 20 will be considered as features for segment alignment.

Table 2. Summary of Proposed Features

Chroma Features	*Tonnetz Features*	*Timbral Features*
Chroma (full range)	Tonnetz (full range)	MFCCs
Chroma[low]	Tonnetz[low]	
Chroma[mid]	Tonnetz[mid]	
Chroma[treble]	Tonnetz[treble]	

"Full-range" Chroma and Tonnetz calculate those features over the entire audio signal, whereas the low, mid, and treble versions only process the signal in the ranges defined in Table 1. MFCCs are the mel-frequency cepstral coefficients, discussed in the text.

The complete set of proposed features is summarized in Table 2. To test their effect on chorus alignment, we used 39 feature sets composed of different combinations of the descriptors Chroma, Tonnetz, and MFCCs. These descriptors are included both individually and with all seven possible combinations, the frequency features split on ranges are also considered individually and combined with both other ranges and other feature types (16 for Chroma and 16 for Tonnetz). Each feature set is formed by the concatenation of its components, and the comparison of how each set performs is tested later in this article.

Distance Metrics

The resemblance between two sequences is measured with a distance metric and the DTW algorithm searches for the optimal path based on these similarities. Thus, the choice of an appropriate distance function is crucial, because depending on the data type and the spread of each feature, some distance functions may reveal similarities more accurately than others. Especially when different kind of features are being concatenated or mixed in various ways, it is important to find a distance metric that

adjusts correctly, because features with wider ranges and higher variance can dominate over the rest.

The Euclidean distance metric

$$d_{\text{euclidean}}(x, y) = \|x - y\| \tag{1}$$

is commonly used for alignment. Nevertheless, it does not consider correlations between data, it is highly sensitive to scaling, and in some cases it will not be the most suitable choice for segment alignment, especially when different features are combined.

We propose that the Mahalanobis distance,

$$d_{\text{mahalanobis}}(x, y) = \sqrt{(x - y)^T C^{-1} (x - y)}, \tag{2}$$

is more appropriate for ensembles of features, because it takes correlations between the data into consideration. The only difference from the Euclidean distance is the inverse covariance matrix (C^{-1}), which acts as a factor decorrelating the data, thereby making the distance computation more robust and reliable. It represents the distance between a point and a distribution, so the covariance matrix should be computed in advance.

Chord Transcription

Modern approaches to ACT tend to be based on deep learning, which can effectively combine the extraction of musically relevant features and a sequence analysis that provides temporal coherence to the chord predictions (McFee and Bello 2017; Jiang et al. 2019; Wu et al. 2019). In comparison with systems that are not integrated, performing each step as an independent process, this kind of architecture allows us to input data with little preprocessing and to directly output class probability for each chord (or each chord component). One of the advantages of these integrated systems is that context, in both time and frequency, can be learned in a data-driven manner.

Network Architecture

Our deep-learning architecture is strongly based on the work presented by McFee and Bello (2017), which implements a CRNN network following an encoder-decoder architecture, using a CQT audio representation as input and producing chord class probabilities as output. The publicly available implementation of the algorithm is known as CREMA (convolutional and recurrent estimators for music analysis). Our approach introduces two variations: First, the number of convolutional layers and their number of filters is increased; second, the structured training targets are different.

In many popular music styles, the bass instrument focuses on the root of the chord, also playing other chord tones such as the fifth. For those styles, it could therefore be effective for a chord detection task to use the bass's pitches as an intermediate target. In the walking bass technique used in jazz, however, the bass does not remain on the root as much and typically plays many nonchordal tones as well, often moving stepwise through the scale. Thus, the bass's pitches cannot be used effectively as an intermediate target.

Instead, we added a beat detector to the intermediate supervision, acting as a binary classifier. It is common that chord transitions are aligned to beats, thus beat onsets should be considered as they encourage the system to transition from one chord to another on beat times. Since the bass is usually present, it facilitates the beat detection task given the fact that in the walking bass technique, notes often occur on all the beats in a bar and not often between beats. Similarly to McFee and Bello (2017), the intermediate outputs are concatenated with the latent features and fed to a bidirectional gated recurrent unit network that acts as a decoder and generates class predictions on a one-to-one manner.

The total loss (objective function for the parameters estimation) is the sum of four different predictions: the root, the chord pitches, the beat onset, and the chord. All of them are considered as classification problems with mutually exclusive classes, except for the chord pitches, which is

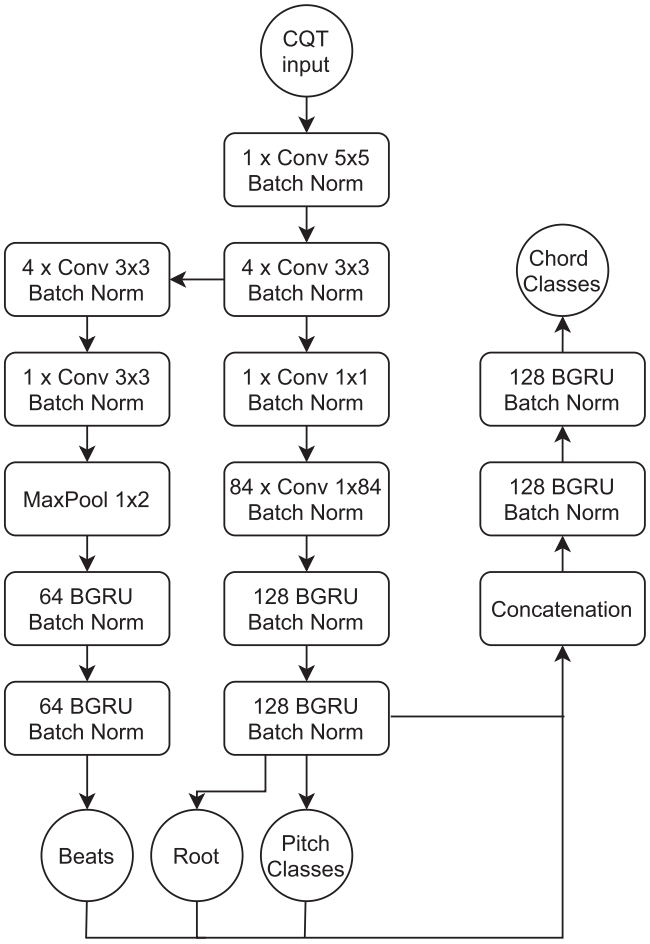


Figure 2. Architecture of the network used for the automatic chord transcriptions of jazz recording. It consists of a convolutional recurrent neural network (CRNN). The input to the CRNN is a representation of the audio using the constant-Q transform (CQT), and the recurrent cells are bidirectional recurrent gated units (BGRU). See text for further details.

addressed as a multilabel task. The cross-entropy loss function is used:

$$H = \sum_{c}^{C} -t_c \log(p_c), \tag{3}$$

where t_c is the true label for class c and p_c represents the prediction score for that class. The total loss is

$$L = \alpha(H_{roots}) + \beta(H_{pitch}) + \gamma(H_{beats}) + \delta(H_{chords}), \tag{4}$$

where each H component is the cross-entropy loss for each subtask and L is the total loss. Each individual component is scaled by one of the factors α, β, γ, or δ. These coefficients are a design choice, because they allow us to increase or decrease the impact of each subtask in the total loss. Thus, they were hand-tuned according to the error ranges of each component, aiming to ensure the predominance of the chord loss over the intermediate outputs.

The full architecture, depicted in Figure 2, has around one million parameters. The training was performed using segments of 5.52 seconds each, chosen randomly for every *epoch* (i.e., each iteration over the training set) from a pool of all training segments. The batch size is 64, using the Adam optimizer (Kingma and Ba 2015), which uses adaptive moment estimation to update the learning rates and is widely used to train deep-learning models.

After each training epoch, validation is performed to track the network's performance, which can be used to prevent overfitting. Our validation was performed on whole performances that were put aside at the beginning of the training, instead of randomly selected segments from the training set. We found out that validating on segments from the same recordings used for training does not reveal overfit, because these segments are too similar.

Chorus Averaging

Every chorus repetition expresses elements of the recording's harmony and the musical content of each should be taken into consideration to achieve a correct chord transcription. Therefore we explore methods to combine all choruses that have been previously aligned. Besides the work presented by Mauch, Noland, and Dixon (2009) there is no precedent, to our knowledge, on how structural segmentation can be integrated to ACT, so we propose the following three alternatives, depicted in Figures 3, 4, and 5.

1. CQT averaging: Originally presented by Mauch, Noland, and Dixon (2009), who mixed the audio content prior to analysis. The resulting averaged chorus can be messy, especially if the recording contains many choruses, as all notes played in a measure across choruses will be considered.

Figure 3. The first proposed method to combine all choruses on the automatic chord transcription: The constant-Q transform inputs are averaged frame by frame, and the results are analyzed by the network.

Figure 4. The second proposed method to combine all choruses on the automatic chord transcription: After encoding each chorus independently, their latent features are averaged and later decoded to predict the chord classes.

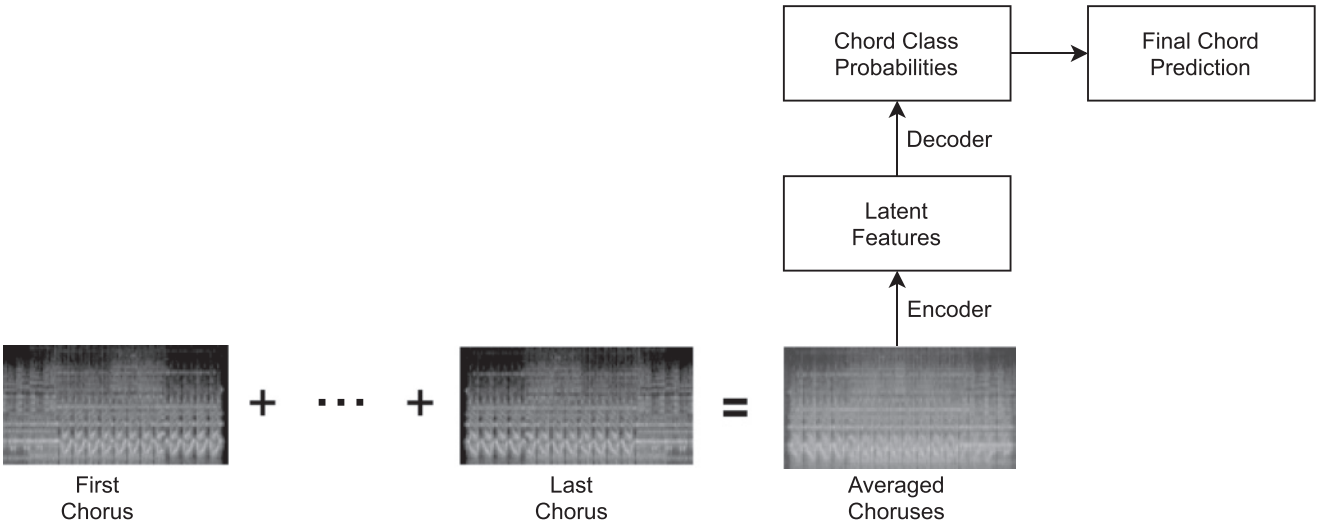


Figure 3.

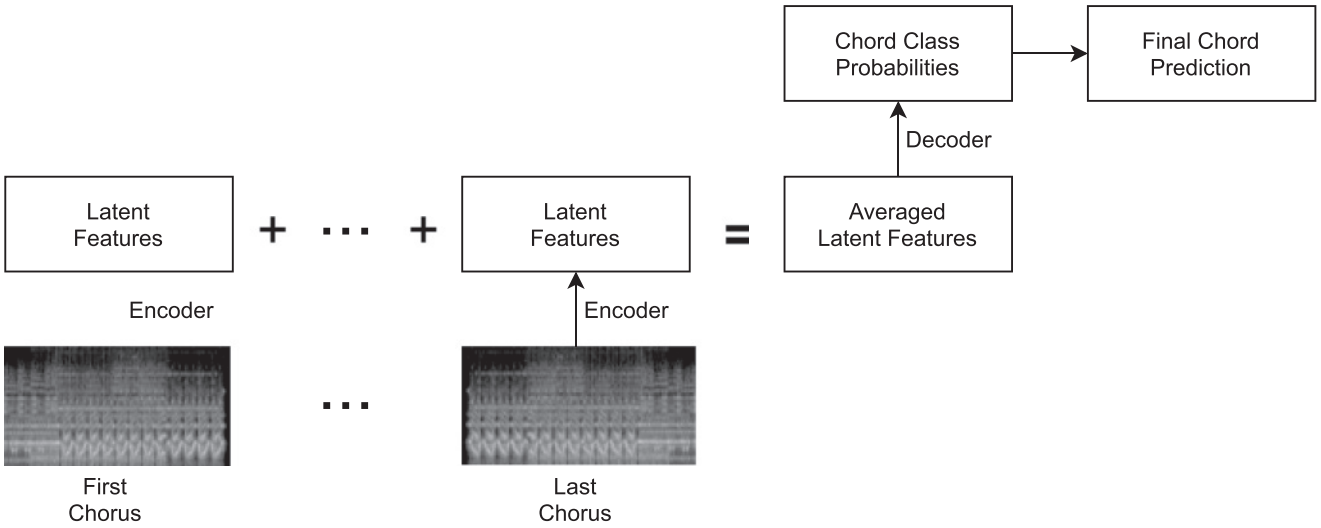
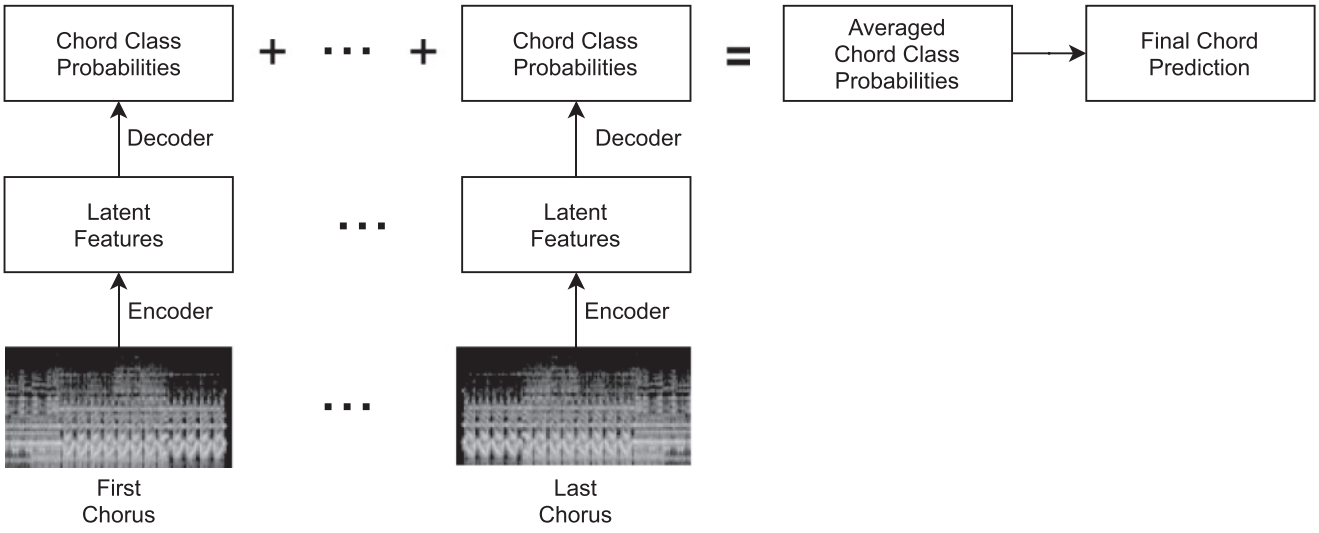


Figure 4.

2. Latent Feature averaging: The latent features, from which the intermediate outputs are obtained, carry most of the melodic and harmonic musical content extracted by the encoder. This content is already refined from the CQT form. As the decoder part of the system inputs the concatenation of the latent features and the intermediate outputs, the latter are also averaged, and only the resulting mixed "latent chorus" is decoded.
3. Prediction averaging: The chord class probabilities are the final output of the network.

Figure 5. The third proposed method to combine all choruses on the automatic chord transcription: Each chorus is encoded and decoded independently, but the output chord-class probabilities are averaged before predicting the most likely chord per frame.



Ideally, the correct chord will have the highest probability, but in other cases it is expected to have relatively high probability. Averaging these probabilities between all choruses repetitions will reduce the impact of chord classes that do not have high probability across all choruses.

Experiments

There was one experiment performed for music synchronization and another for chord transcription. Both were performed on the same database, using audio recordings having a sample rate of 44,100 Hz. For both experiments, the CQT is computed with one bin per semitone spanning a range from 32.7 to 3,951.1 Hz (from C1 to B7), resulting in 84 frequency bins and using a hop length of 46 msec. As preprocessing, all recordings in which the pitch diverged slightly from the A440 reference were corrected, without changing the original key. Applying this tuning correction ensured that every bin in the CQT represented the same frequency content for all recordings.

Database

The Jazz Audio-Aligned Harmony (JAAH) dataset, presented by Eremenko et al. (2018), consists of 113 different jazz recordings with transcribed annotations. The annotations comprise the complete set of beats, indicating temporal location in seconds, the beat-aligned chord sequence, and labels for each structural segment ("intro," "sax solo," etc).

The JAAH includes a wide range of jazz subgenres, ranging from its origins up to the beginning of modal and free jazz. Nevertheless, Eremenko et al. (2018) selected the tracks for the database homogeneously, and there is no bias on period or subgenre. Most of the tracks follow the standard structure in which the chorus is repeated multiple times. Some recordings did not exhibit this structure, however, because they were special arrangements of a composition in which measures may have been added to the chorus or omitted. Those recordings were discarded, nevertheless the database remains homogeneous. Only the chorus sections were used for experimentation and other segments (introduction, bridge, "outro," etc.) were not considered. Also, some recordings are different renditions of the same song but are sufficiently different to be

considered independent samples, and so were kept. For the alignment experiment 86 files were considered. For chord transcription, 78 of the files were used for training, 2 for validation, and 6 for testing; equivalent to 4 hours, 11 minutes, and 15 minutes, respectively.

The annotated chords were not identical between choruses, because the manual transcriptions included variations that might have been played. For example, when the main melody is performed, the chords can be slightly different to those used in the solos, so the annotated harmonic sequences may differ. These details were not desired in our general harmonic analysis, even if they can be useful for other tasks in harmonic analysis. To address this problem, we compared the chords for each beat across all choruses, and chose the most frequently occurring chord. This process aimed to mimic what a musician would do if asked to transcribe the chord sequence of a jazz performance to a single chorus lead sheet.

Chord distributions are highly biased to the most frequent keys in jazz recording; the ones easier to play in instruments like saxophone or trumpet (F, E-flat, B-flat, C, etc.). To avoid bias towards those keys in the ACT experiment, we included data augmentation consisting of shifting each recordings's pitch by an integral number of semitones, ranging from -5 to $+6$, thereby including all twelve possible transpositions. This resulted in $86 \times 12 = 1{,}032$ tracks.

Chord Vocabulary

For the ACT experiment, the chord vocabulary needed to be chosen carefully, because plain triads are almost never used in jazz, and extension notes can be added without changing the chord's function. We used the chord vocabulary developed by Eremenko et al. (2018), who provide insights regarding how chords should be classified in jazz music, resulting in five chord classes that play a different harmonic function. These classes are major (maj), minor (min), dominant seventh (7, written here as a superscript), half-diminished seventh (hdim^7), and diminished seventh (dim). Additionally, a no-chord symbol is considered (N), resulting in 61 classes (five chord classes for each of the twelve pitches, plus the no-chord symbol).

Evaluation Metrics

As two different experiments were conducted, each needs its own evaluation metrics, which will be described in the following.

Evaluation of Chorus Alignment

The true alignment lives on a continuous time domain and can be approximated on a discrete time domain (in our case one sample every 46 msec), resulting in the best possible alignment (ground truth) given by matching beat onsets. On the other hand, the optimal path found by the DTW algorithm is discrete.

The evaluation of music synchronization systems can be understood as a geometric problem, comparing the distance of the optimal path to the ground truth path on a 2-D plane. To measure this error, the mean absolute error (MAE) was used, which is defined as

$$\text{MAE} = \frac{\sum_{i=1}^{N} |y_i - x_i|}{N}, \tag{5}$$

where

y_i is a point on the estimated path,
x_i is its closest point from the ground truth path, and
N is the length of the optimal path.

This expression was chosen mainly because it is measured in temporal units, making the interpretation simple and intuitive.

Because error is measured on continuous time, there will be a small error induced by discretization, meaning that a perfect alignment will not have zero error. With a sampling rate of 44,100 Hz, there is a lower threshold on time resolution equal to 22.7 μsec; everything below that does not have real meaning and is rounded to zero.

Evaluation of Chord Transcription

Measuring the performance of ACT systems is complex, because there are many musically meaningful elements that must be considered. Chord classes cannot be considered as independent, because some of them share some pitch content—for example C and C^7, or Dmin and F. To overcome this issue, five evaluation metrics commonly used in ACT research are computed, each of which describes a particular harmonic relation between chords.

Chords are evaluated element-wise, because both ground truth and predicted sequences have the same length. Some standard evaluation metrics are implemented by Raffel et al. (2014) on the Python library MIR_EVAL, but not all of them are appropriate to the chord classes used in this work. Although some of those metrics are not particularly well suited for jazz chord transcription, they are commonly used in ACT systems, so they are included to give comparable results to other research. In particular, we use the accuracy (percentage of correct classifications) and the following subset of evaluation metrics:

1. Root: Chords match if they have the same root.
2. Major/minor: Chords match if they share root and chord type, providing it is major or minor. Other chord classes are not considered.
3. Third: Chords match if they share the root and third. For example, chords Emin and Edim match.
4. Triad: Chords match if they have the same basic triad. For example, chords E^7 and E match, but Emin and Edim do not.
5. MIREX: Chords match if they share at least three pitch classes. For example, F^7 and Adim match. This evaluation metric is used on the Music Information Retrieval Exchange (MIREX) competitions on chord transcription.

The beat detection problem uses standard detection evaluation metrics: Precision, recall, and F-measure. As the hop length is 46 msec, a tolerance of ±25 msec forces the frames to match exactly, which considerably reduces the performance on those three metrics. On the other hand, ±50 msec correspond to a tolerance of ±1 frame, allowing more flexibility. Both results will be considered, to show the sensibility of the detection evaluation metrics in this problem.

Results and Discussion

In this section, the results of the alignment and ACT experiments are presented separately. Nevertheless, the best alignment method obtained was used to synchronize the choruses for the ACT experiment.

Chorus Alignment Results

The inverse covariance matrix required to calculate the Mahalanobis distance is computed from the two aligned sequences, meaning that a new matrix is obtained for every possible combination of two choruses. This process takes about 1 msec for each pair running on an 3.70 GHz Intel i7-8700K CPU.

For the sake of clarity, only the top three results, and the average of all feature sets, are displayed for each distance function in Table 3. Both Chroma and Tonnetz features that are obtained on different frequency ranges are indicated by square brackets, whereas the versions on the whole frequency range are displayed without brackets. All results are in seconds.

It is possible to compare the average performance of two mutually exclusive subsets: one with all the combinations including a specific feature and another not containing that feature. For example, the ones using MFCCs and those without, showing the impact of the addition of the MFCCs to a feature set. These comparative results are presented in Table 4, which shows the improvement of the first subset (before the slash mark "/") over the second subset (after the slash).

The Mahalanobis distance achieved considerably better results than Euclidean distance, showing that the inverse covariance matrix plays a major role when comparing sequences of similarity features.

The Tonnetz features were proposed as an alternative to the Chromagram, as the former allow modeling musical relations between notes. Despite this characteristic, Chroma yielded better results on

Table 3. Chorus Alignment Results

Distance Function	*Features*	*MAE*
Euclidean	Average of all feature sets	331
	Chroma[bass+mid+treble] + Tonnetz + MFCCs	129
	Chroma[bass+mid+treble] + Tonnetz	132
	Chroma[bass+mid+treble] + MFCCs	134
Mahalanobis	Average of all feature sets	183
	Chroma[bass+mid+treble] + MFCCs	51
	Chroma[bass+mid] + MFCCs	54
	Chroma + MFCCs	63

MAE is the mean absolute error and given in milliseconds.

Table 4. Percentage of Improvement for Mutually Exclusive Feature Subsets

	% of Improvement	
Subset Comparation	*Euclidean*	*Mahalanobis*
MFCCs: With/Without	17.3	67.1
Chroma[bass+mid+treble] / full range	46.9	38.0
Chroma[bass+mid+treble] / [bass]	49.2	45.3
Chroma[bass+mid+treble] / [mid]	53.5	46.3
Chroma[bass+mid+treble] / [bass+mid]	20.7	15.3
Tonnetz[bass+mid+treble] / full range	14.4	15.4
Tonnetz[bass+mid+treble] / [bass]	31.3	39.0
Tonnetz[bass+mid+treble] / [mid]	32.3	39.1
Tonnetz[bass+mid+treble] / [bass+mid]	13.0	12.9
Chroma/Tonnetz	17.9	37.3

average, and the difference was higher when using the Mahalanobis metrics, as can be seen in the last row of Table 4.

Not all frequency ranges carry the same amount of musically relevant information for this task, as has been shown by the "[bass+mid+treble]" version achieving 46.9% lower error than the "Normal" version when using the Euclidean metric and 38.0% using Mahalanobis. Comparing the three-range features against only bass and only mid-range led to similar improvement, and the treble frequency range is the one that showed the smallest improvement when added to the other two, as can be seen in the comparison between "[bass+mid+treble]" and "[bass+mid]." In the case of the Tonnetz features the improvements achieved with Euclidean metric are similar or lower than for the Mahalanobis metric. Both bass and middle ranges contribute similarly, and treble had a smaller contribution.

Features including MFCCs are present among the feature sets with highest scores, especially when using Mahalanobis distance function. As can be seen in Table 4, the results including MFCCs are 67.1% better than those without. This improvement is considerably smaller using Euclidean distance (only 17.3%), suggesting that MFCC features contribute the most when their relations with melodic and harmonic features can be considered.

Table 5. Chord Evaluation Metrics

	Accuracy	*Root*	*Major/Minor*	*Third*	*Triad*	*MIREX*
Not averaged	0.52	0.59	0.59	0.58	0.58	0.60
CREMA	0.36	0.49	0.47	0.46	0.46	0.48
CQT averaging	0.60	0.66	0.67	0.64	0.64	0.67
Latent feature averaging	0.62	0.69	0.68	0.67	0.66	0.68
Chord prediction averaging	**0.64**	**0.70**	**0.71**	**0.68**	**0.68**	**0.71**

The highest values in each column are in **boldface**.

Table 6. Beat Detection Metrics

Tolerance	*Averaging*	*Precision*	*Recall*	*F-measure*
50 msec	Not averaged	0.89	0.87	0.88
	CQT	**0.96**	**0.94**	**0.95**
	Latent Features	0.95	**0.94**	**0.95**
25 msec	Not averaged	0.51	0.50	0.50
	CQT	0.58	0.57	0.58
	Latent Features	**0.63**	**0.62**	**0.63**

The highest values in each column are in **boldface**.

Chord Transcription Results

The network was trained on a single Nvidia GeForce GTX 1080 GPU for about 15 epochs before it started overfitting. Each epoch takes around 30 minutes, which is quite low compared to many deep-learning models.

We compare our method to the publicly available implementation of CREMA, which we ran only on our test database. These results, shown in Table 5, are split into two parts: The upper two rows show performance for all choruses without averaging, as ACT systems usually do. The final three rows show the methods that combine segments, thus transcribing a chord sequence for only one calculated "chorus," which is the average across choruses.

The beat detection results are shown in Table 6. It is important to notice that, except for the tests that didn't use averaging, these beats correspond to the beat sequence of all choruses reduced to one chorus and not of the whole performance.

Lead sheet–like representations were obtained from the chord predictions, as can be seen in Figures 6 and 7. The network's outputs are beat predictions an a chord list, which is reshaped to match the recording's bars, and chord transitions that do not fall on a beat are discarded. The number of bars and the metric are inputs and are not automatically calculated by the system.

As can be seen on Table 5, our system obtains considerably higher performance than CREMA (more than 10% for each metric), when choruses are not averaged. The performance improvement can be explained by the fact that our model is trained and tested on a genre-specific database and probably will not perform properly on other genres, and CREMA might be able to better generalize to more genres. These results show that a system that has been trained on a large dataset, but not focused on jazz, does not generalize correctly to this genre.

The highest score was obtained when averaging chord probabilities for each chorus (final row in Table 5) and the latent features averaging (penultimate row) scored second. This suggests that mixing the sequences after being processed by a trained system leads to higher performance than combining raw inputs. The latent features are obtained by the encoder and the chord probabilities by the encoder-decoder, showing that when more processing is added the harmonic and melodic content seems to be refined, at least for chord transcription.

In general, the system correctly detects chord changes, especially on coarse time quantization like beats. Nevertheless, around 29% of the chords are misclassified (using the MIREX evaluation metric)

Figure 6. Lead sheet–like representation obtained for Django Reinhart's "Minor Swing." At each bar, ground truth chords are shown above and predicted chords below. (Note: The underscore character is used to indicate minor chords.)

and it is found that some of these errors were caused by chord transitions that were advanced or delayed by at least one beat. This effect is found even when the two chords involved are not similar, implying that the error does not depend directly on their musical content. Despite including a beat detector, the network cannot always reliably capture the temporal frame where a chord transitions. This issue could be alleviated by postprocessing the transcribed chords using hidden Markov models, as done by McFee and Bello (2017), or using language models to correct shifted chord transitions.

The beat detector seems to benefit from the chorus combination techniques, nevertheless all three methods have similar results. The system is highly sensitive to the temporal tolerance and the results increase by more than 30% when using ±50 msec instead of ±25 msec. A hop length of 46 msec represents a coarse temporal resolution for onset detection tasks, but is appropriate for chord transcription.

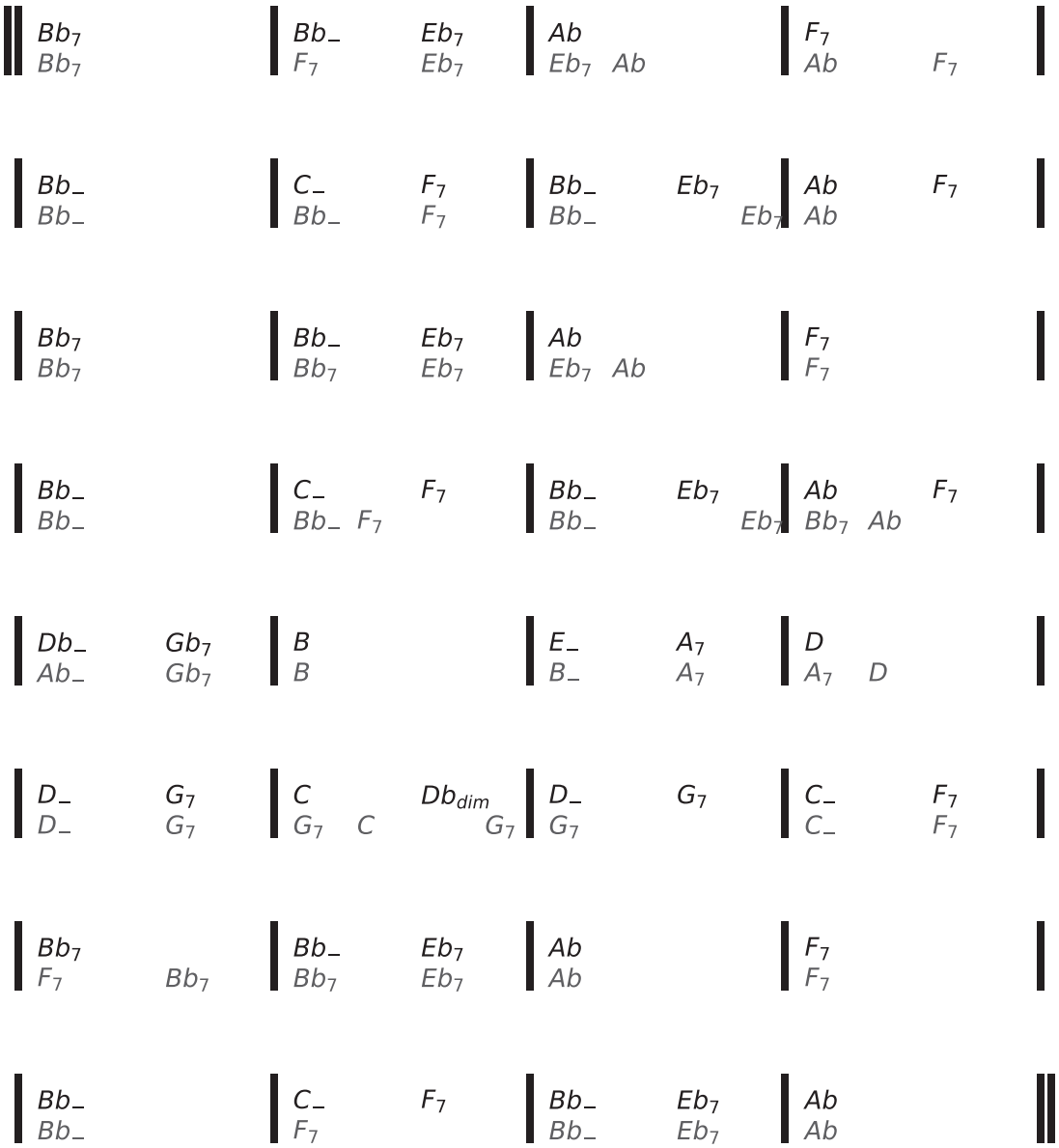
Conclusions

We presented a novel approach for automatic chord transcription of jazz music that performs an exhaustive analysis of the audio incorporating the musical attributes of the genre. We focused on specific aspects of jazz that were exploited, in particular the repetitive nature of a fundamental segment called the chorus. We presented a study on chorus alignment, where insights were obtained regarding how this task should be addressed for jazz. Later, we used the best method obtained to align all choruses on a recording and performed chord transcription combining the musical content from each chorus.

The segment-alignment study focused on the features input into the dynamic time-warping algorithm, and how distance metrics exploited their similarities. We found that using large feature sets, describing musical aspects like pitch and timbre, are considerably better at unveiling similarities than single features like Chromagram, especially when using the Mahalanobis distance metric.

Regarding the chord transcription stage, we merged a beat detector and a chord predictor into a single system, allowing us to have better beat-synchronized chord sequences, which discouraged chord transitions that were not on a beat. We explored three options to combine the harmonic content of each chorus and we concluded that averaging the chord probabilities provides the best results. This suggest that combining sequences after being processed by a trained system leads to better performance than mixing the raw inputs or the intermediate latent features. Our method achieves better performance on jazz music than other systems trained on larger databases not specific to jazz.

Figure 7. Lead sheet–like representation obtained for Jimmy Giuffre's "Four Brothers." At each bar, ground truth chords are shown above and predicted chords below. (Note: The underscore character is used to indicate minor chords.)



Future Work

The JAAH database, which is used train the ACT system, is moderately large and sufficient for many musical research tasks. Nevertheless, is not enough to train a robust chord transcription system without overfitting. In our case, the system was trained on over 15 epochs before validation loss started to raise. We believe that a larger jazz database could significantly improve the results, especially for chord classes that have little presence, like diminished or half-diminished sevenths. Also the chord vocabulary could be expanded to include other chord classes, such as augmented triads (currently

considered as major) or suspended fourths ("sus" chords).

Even if evaluation metrics were carefully chosen and the chord dictionary is well suited for jazz music, there are aspects regarding how information is notated and how performance is measured that can be improved for this musical genre. As can be seen in Figure 6, on the 14th measure the annotated chord is $B\flat^7$ and the predicted chord is E^7. Under all current evaluation metrics the predicted chord is incorrect, but the two chords are closely related because they are tritone substitutions (widely used in jazz, as noted earlier) and there should be at least one metric that takes this aspect into account. Even if these errors are not very frequent, they have a meaningful musical interpretation and should be addressed.

In our work, the musicological perspective was not included with the adequate depth. This aspect was not further explored because it is beyond the scope of our current research, but the next steps of this work should include an exhaustive literature review on jazz harmony and probably needs supervision from a jazz musician, musicologist, or music historian. A better understanding from the musical perspective may lead to changes in the way an ACT system is designed, implemented, and evaluated.

Acknowledgments

The research reported in this article is supported by the Agencia Nacional de Investigación y Desarrollo de Chile through Fondo Nacional de Desarrollo Científico y Tecnológico, Project No. 1201551.

References

Bello, J. P., and J. Pickens. 2005. "A Robust Mid-Level Representation for Harmonic Content in Music Signals." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 304–311.

Boulanger-Lewandowski, N., Y. Bengio, and P. Vincent. 2013. "Audio Chord Recognition with Recurrent Neural Networks." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 3335–3340.

Brown, J. 1991. "Calculation of a Constant Q Spectral Transform." *Journal of the Acoustical Society of America* 89:425–434.

Duan, Z., and B. Pardo. 2011. "Aligning Semi-Improvised Music Audio with Its Lead Sheet." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 513–518.

Eremenko, V., et al. 2018. "Audio-Aligned Jazz Harmony Dataset for Automatic Chord Transcription and Corpus-Based Research." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 483–490.

Ewert, S., M. Müller, and P. Grosche. 2009. "High Resolution Audio Synchronization Using Chroma Onset Features." In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1869–1872.

Fujishima, T. 1999. "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music." In *Proceedings of the International Computer Music Conference*, pp. 464–467.

Harte, C., M. Sandler, and M. Gasser. 2006. "Detecting Harmonic Change in Musical Audio." In *Proceedings of the ACM Workshop on Audio and Music Computing Multimedia*, pp. 21–26.

Humphrey, E. J., and J. P. Bello. 2012. "Rethinking Automatic Chord Recognition with Convolutional Neural Networks." In *Proceedings of the International Conference on Machine Learning and Applications*, vol. 2, pp. 357–362.

Humphrey, E. J., and J. P. Bello. 2015. "Four Timely Insights on Automatic Chord Estimation." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 673–679.

Izmirli, Ö., and R. B. Dannenberg. 2010. "Understanding Features and Distance Functions for Music Sequence Alignment." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 411–416.

Jiang, J., et al. 2019. "Large-Vocabulary Chord Transcription via Chord Structure Decomposition." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 644–651.

Kingma, D. P., and J. L. Ba. 2015. "Adam: A Method for Stochastic Optimization." In *Proceedings of the International Conference on Learning Representation*. Available online at arxiv.org/pdf/1412.6980.pdf. Accessed August 2021.

Korzeniowski, F., and G. Widmer. 2016. "A Fully Convolutional Deep Auditory Model for Musical Chord

Recognition." In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6.
Maezawa, A., et al. 2014. "Bayesian Audio Alignment Based on a Unified Model of Music Composition and Performance." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 233–238.
Mauch, M. 2010. "Automatic Chord Transcription from Audio Using Computational Models of Musical Context." PhD dissertation, Queen Mary University of London.
Mauch, M., K. C. Noland, and S. Dixon. 2009. "Using Musical Structure to Enhance Automatic Chord Transcription." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 231–236.
McFee, B., and J. P. Bello. 2017. "Structured Training for Large-Vocabulary Chord Recognition." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 188–194.
McVicar, M., et al. 2014. "Automatic Chord Estimation from Audio: A Review of the State of the Art." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22:556–575.
Müller, M., S. Ewert, and S. Kreuzer. 2009. "Making Chroma Features More Robust to Timbre Changes." In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1877–1880.
Ni, Y., et al. 2012. "An End-to-End Machine Learning System for Harmonic Analysis of Music." *IEEE Transactions on Audio, Speech, and Language Processing* 20:1771–1783.
Ni, Y., et al. 2013. "Understanding Effects of Subjectivity in Measuring Chord Estimation Accuracy." *IEEE Transactions on Audio, Speech, and Language Processing* 21:2607–2615.
Pachet, F., J. Suzda, and D. Martínez. 2013. "A Comprehensive Online Database of Machine-Readable Lead-Sheets for Jazz Standards." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 275–280.
Pauwels, J., et al. 2019. "20 Years of Automatic Chord Recognition from Audio." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 54–63.
Raffel, C., et al. 2014. "MIR EVAL: A Transparent Implementation of Common MIR Metrics." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 367–372.
Ryynänen, M., and A. Klapuri. 2007. "Automatic Bass Line Transcription from Streaming Polyphonic Audio." *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07* 4:IV-1437–IV-1440.
Sheh, A., and D. P. W. Ellis. 2003. "Chord Segmentation and Recognition Using EM-Trained Hidden Markov Models." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 183–189.
Sumi, K., et al. 2008. "Automatic Chord Recognition Based on Probabilistic Integration of Chord Transition and Bass Pitch Estimation." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 39–44.
Wang, S., S. Ewert, and S. Dixon. 2014. "Robust Joint Alignment of Multiple Versions of a Piece of Music." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 83–88.
Wu, Y., T. Carsault, and K. Yoshii. 2019. "Automatic Chord Estimation Based on a Frame-Wise Convolutional Recurrent Neural Network with Non-Aligned Annotations." In *Proceedings of the European Signal Processing Conference*, pp. 1–5.