END-TO-END MELODY NOTE TRANSCRIPTION BASED ON A BEAT-SYNCHRONOUS ATTENTION MECHANISM

Ryo Nishikimi[1] *Eita Nakamura*[1] *Masataka Goto*[2] *Kazuyoshi Yoshii*[1]

[1]Graduate School of Informatics, Kyoto University, Japan
[2]National Institute of Advanced Industrial Science and Technology (AIST), Japan
{nishikimi, enakamura, yoshii}@sap.ist.i.kyoto-u.ac.jp, m.goto@aist.go.jp

ABSTRACT

This paper describes an end-to-end audio-to-symbolic singing transcription method for mixtures of vocal and accompaniment parts. Given audio signals with *non-aligned* melody scores, we aim to train a recurrent neural network that takes as input a magnitude spectrogram and outputs a sequence of melody notes represented by pairs of pitches and note values (durations). A promising approach to such sequence-to-sequence learning (joint input-to-output alignment and mapping) is to use an encoder-decoder model with an attention mechanism. This approach, however, cannot be used straightforwardly for singing transcription because a *note-level* decoder fails to estimate note values from latent representations obtained by a *frame-level* encoder that is good at extracting instantaneous features, but poor at extracting temporal features. To solve this problem, we focus on tatums instead of notes as output units and propose a *tatum-level* decoder that sequentially outputs tatum-level score segments represented by note pitches, note onset frags, and beat and downbeat flags. We then propose a beat-synchronous attention mechanism constrained in order to monotonically align tatum-level scores with input audio signals with a steady increment. The experimental results showed that the proposed method can be trained successfully from non-aligned data thanks to the beat-synchronous attention mechanism.

Index Terms— Automatic singing transcription, end-to-end learning, sequence-to-sequence learning, encoder-decoder recurrent neural networks, attention mechanism

1. INTRODUCTION

Automatic singing transcription (AST), *i.e.*, estimating a sequence of musical notes corresponding to a sung melody from a music audio signal, is a challenging task that generally consists of multiple sequential sub-tasks. Singing voice separation [1, 2] and vocal F0 estimation (a.k.a. melody extraction) [3–6] are used for estimating an F0 trajectory of singing voice, which is then quantized in the frequency and temporal directions to estimate the pitches, onset times, and durations of notes by using note-tracking methods [7–10] and rhythm transcription methods [11, 12]. This cascading process, however, suffers from the error propagation problem.

Since AST is a typical sequence-to-sequence task that maps a sequence of acoustic features to a sequence of musical note symbols, the end-to-end approach based on a neural encoder-decoder model with an attention mechanism is considered to be promising in theory. In general, an encoder is used for converting an input sequence into a sequence of latent representations of the same length. A decoder is then used for sequentially predicting an output sequence of an appropriate length from the latent sequence while associating each output symbol with input symbols (frames) in the attention mechanism. While this approach has been investigated intensively and made a big success in machine translation [13, 14] and automatic speech recognition (AST) [15–17], only a few attempts have been conducted for automatic music transcription [18].

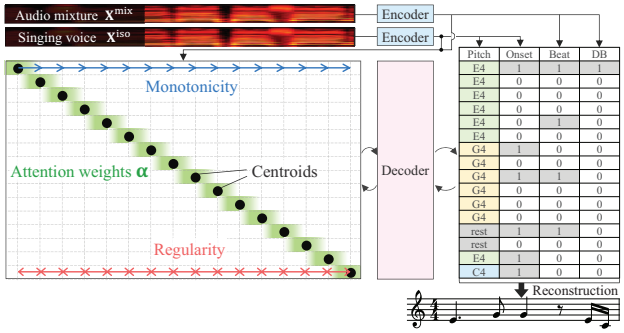

Figure 1: The proposed neural encoder-decoder model with a beat-synchronous attention mechanism for end-to-end singing transcription. DB = 'downbeat'. New loss functions for the centroids of attention weights are introduced to align them with equally-spanned beat times.

If we limit our focus to monophonic music transcription, the fundamental difficulty of attention-based AST lies in estimation of temporal information of musical notes (metrical positions of note onsets and note values). Previously, a standard attention-based model consisting of a frame-level encoder and a *note-level* decoder was used for AST [18]. This model was found to be able to estimate note pitches, but often failed to estimate note values. This is because that the encoder is good at extracting instantaneous features (pitch and timbral information) from an input sequence, but poor at extracting temporal features (duration information). Even long short-term memory networks (LSTMs) cannot propagate temporal information through several tens or hundreds of frames that a note duration would have.

To solve this problem, we consider tatums (16th-note-level beat times) as output symbols instead of musical notes and propose a new attention-based model consisting of two frame-level encoders followed by a *tatum-level* decoder (Fig. 1). The two encoders independently extract latent representations about notes and beats from singing signals (assumed to be separated in advance) and music signals, respectively. These representations are used jointly for calculating the attention weights for an output symbol at each step. The decoder sequentially predicts output symbols, *i.e.*, tatum-level

This work was supported in part by JST ACCEL No. JPMJAC1602, JSPS KAKENHI No. 16H01744, No. 19H04137, and No. 19K20340

score segments consisting of note pitches, note onset flags, and beat and downbeat flags. This architecture is thus considered to be able to make pitch- and beat-aware accurate input-output alignment by leveraging the metrical structures of notes and beats.

We further propose a beat-synchronous attention mechanism that imposes constraints on the attention weights in terms of *monotonicity* and *regularity*. Since popular songs typically have steady tempo and regular beats, these constraints guide together the attention centroids of output symbols to line up in ascending order with an almost equal interval. Conventional methods implement the monotonicity constraint by modifying network architectures [19] or designing a special architecture of calculating attention weights [20], whereas we implement both monotonicity and regularity constraints in the loss functions that are minimized jointly with the cross-entropy loss for output symbols.

The main contribution of this paper is to propose a new attention model with a tatum-level decoder for popular music having regular metrical structure. Our model can jointly perform note estimation with beat and downbeat tracking in a unified framework. We experimentally investigate the effectiveness of the monotonicity and regularity constraints.

2. RELATED WORK

This section reviews related work on automatic music transcription.

2.1. Piano-roll estimation

Many studies have attempted to estimate a piano-roll representation in which pitches are quantized in semitones but onset times and durations are not quantized and are represented in seconds (or frame indices) [7–10, 21, 22]. The piano-roll estimation for singing voice is usually performed by note tracking (*i.e.*, pitch quantization and note region detection) for F0 trajectories estimated in advance. Note tracking methods are usually based on hand-crafted rules and filters [7, 8] or hidden Markov model (HMM) [9, 10]. Some note tracking methods for other musical instruments (*e.g.*, piano) directly deal with not F0s but spectrograms because pitches of the instruments are stabler and onset times are clearer than those of singing voice. Spectrogram factorization techniques like probabilistic latent component analysis (PLCA) [21] are employed to estimate discrete pitches of each time frame, followed by note tracking based on HMMs. Hawthorne *et al.* [22] proposed a method based on LSTMs that estimate semitone-level pitches and onset frames directly from an input spectrogram.

2.2. Musical score transcription

There are several attempts to estimate not a piano roll but a complete score [12, 18, 23–28]. Nakamura *et al.* [12] proposed a rhythm transcription method based on a metrical HMM [23, 24] that takes the piano-roll representation including performance fluctuations in the time direction and outputs quantized onset times and note values in tatums. Nishikimi *et al.* [25] and Nakamura *et al.* [26] formulated hidden semi-Markov models (HSMMs) that quantize an F0 trajectory of singing voice to estimate semitone-level pitches, tatum-level onset times, and note values by using beat times estimated in advance. End-to-end approaches to audio-to-score transcription based on deep neural networks (DNNs) have recently been studied. Carvalho *et al.* [27] proposed a method based on the sequence-to-sequence model [29] that estimates the symbols of Lilypond format [30] from features extracted from an audio signal of synthesized piano sound by using a one-dimensional convolutional neural network (CNN). Román *et al.* [28] proposed a method based on connectionist temporal classification (CTC) [31] that estimates the sequence of music symbols such as keys, pitches, note values, and time signatures from the magnitude spectrograms of synthesized piano signals. Nishikimi *et al.* [18] estimated a sequence of pitches and note values from an isolated solo singing voice based on an encoder-decoder model with a weakly-supervised attention mechanism. The mechanism guides the attention weights into ideal values by using ground-truth onset times of musical notes and enables the model to estimate pitches and note values correctly from vague onset times of singing voice. One difference between the proposed method and the conventional one [18] is the joint estimation of notes (pitches and onsets) and metrical structure (beats and downbeats). Another difference lies in the new loss functions based on the unsupervised constraints of attention weights.

3. PROPOSED METHOD

The proposed method of AST is based on an encoder-decoder model with an attention mechanism (Fig. 1). The proposed model has two encoders that take as inputs singing and mixture spectrograms, respectively, and output latent vectors for each frame. These latent vectors are then input to a decoder with a beat-synchronous attention mechanism to output tatum-level score segments.

3.1. Problem specification

The input is a music audio signal and the output is a symbolic score of the vocal part. Let T, F, and N be the number of time frames, that of frequency bins, and that of the 16th-note-level time indices, respectively. The inputs for the proposed network are the mel-scale spectrogram of an isolated singing voice $\mathbf{X}^{\mathrm{iso}} = [\mathbf{x}_1^{\mathrm{iso}}, \ldots, \mathbf{x}_T^{\mathrm{iso}}] \in \mathbb{R}_+^{F\times T}$ and that of a mixture of singing voice and accompaniment $\mathbf{X}^{\mathrm{mix}} = [\mathbf{x}_1^{\mathrm{mix}}, \ldots, \mathbf{x}_T^{\mathrm{mix}}] \in \mathbb{R}_+^{F\times T}$, where $\mathbf{x}_t^{\mathrm{iso}}$ and $\mathbf{x}_t^{\mathrm{mix}}$ indicate the mel-scale spectra at frame t of isolated and mixture signals. The isolated singing voice $\mathbf{X}^{\mathrm{iso}}$ is used as an input to simplify the problem because it is difficult to directly estimate the pitch and onset flag from the mixture signal that includes accompaniment sounds. The output of the network is a sequence of symbols $\mathbf{Y} = [y_1, \ldots, y_N]$. Each symbol $y_n = (p_n, o_n, b_n, d_n)_n$ consists of four symbols: a semitone-level pitch $p_n \in \{1, \ldots, K, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{p}}$, where K represents the size of the pitch vocabulary (including the rest), an onset flag $o_n \in \{0, 1, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{o}}$, a beat flag $b_n \in \{0, 1, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{b}}$, and a downbeat flag $d_n \in \{0, 1, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{d}}$. We can reconstruct a score from the information contained in $\mathbf{Y}$. The special elements, $\langle \mathtt{sos} \rangle$ and $\langle \mathtt{eos} \rangle$, in the vocabulary of each symbol represent the start and end of an output sequence.

3.2. Frame-level encoders

The proposed model has two encoders for $\mathbf{X}^{\mathrm{iso}}$ and $\mathbf{X}^{\mathrm{mix}}$. The encoders transform the spectrograms into sequences of intermediate representation vectors $\mathbf{H}^{\mathrm{iso}} = [\mathbf{h}_1^{\mathrm{iso}}, \ldots, \mathbf{h}_T^{\mathrm{iso}}] \in \mathbb{R}^{E\times T}$ and $\mathbf{H}^{\mathrm{mix}} = [\mathbf{h}_1^{\mathrm{mix}}, \ldots, \mathbf{h}_T^{\mathrm{mix}}] \in \mathbb{R}^{E\times T}$, where E is the dimension of the intermediate representation vectors. Since the length of the input spectrogram is variable, we use as each of the encoders a recurrent neural network (RNN), specifically a multi-layer bidirectional LSTM network.

3.3. Tatum-level decoder with an attention mechanism

The decoder predicts a sequence $\mathbf{Y}$ from the latent vectors $\mathbf{H} = \mathbf{h}_{1:T} \in \mathbb{R}^{2E\times T}$, where $\mathbf{h}_t$ is a concatenation of the intermediate

vectors $\mathbf{h}_t^{\mathrm{iso}}$ and $\mathbf{h}_t^{\mathrm{mix}}$. The decoder consists of a unidirectional LSTM and is defined as follows:

$$\boldsymbol{\alpha}_n = \mathrm{Attend}(\mathbf{s}_{n-1}, \boldsymbol{\alpha}_{n-1}, \mathbf{H}), \tag{1}$$

$$\mathbf{g}_n = \sum_{t=1}^{T} \alpha_{nt} \mathbf{h}_t, \tag{2}$$

$$y_n = \mathrm{Generate}(\mathbf{s}_{n-1}, \mathbf{g}_n), \tag{3}$$

$$\mathbf{s}_n = \mathrm{Recurrency}(\mathbf{s}_{n-1}, \mathbf{g}_n, y_n), \tag{4}$$

where $\boldsymbol{\alpha}_n \in \mathbb{R}^T$ is a set of attention weights, $\mathbf{s}_n \in \mathbb{R}^D$ denotes the n-th hidden state of the decoder, and the functions Attend, Generate, and Recurrency are explained in the following .

(1) and (2) represent the attention mechanism. $\boldsymbol{\alpha}_n \in \mathbb{R}^T$ is a vector of normalized probabilities representing the degrees of relevance between the latent vectors $\mathbf{H}$ and each hidden state $\mathbf{s}_n$. Each element of $\boldsymbol{\alpha}_n$ is calculated as follows:

$$\alpha_{nt} = \frac{\exp(e_{nt})}{\sum_{t'=1}^{T} \exp(e_{nt'})}, \tag{5}$$

$$e_{nt} = \mathrm{Score}(\mathbf{s}_{n-1}, \mathbf{h}_t, \boldsymbol{\alpha}_{n-1}), \tag{6}$$

where Score is a function that calculates a raw weight. We calculate a shared matrix of attention weights $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{1:N} \in \mathbb{R}^{N \times T}$ from $\mathbf{H}^{\mathrm{iso}}$ and $\mathbf{H}^{\mathrm{mix}}$ so that the attention mechanism can put emphasis on the intermediate representations that are important in terms of both note and beat structures. We use as the Score function a convolutional function [15] given by

$$\mathbf{f}_n = \mathbf{F} * \boldsymbol{\alpha}_{n-1}, \tag{7}$$

$$e_{nt} = \mathbf{w}^\top \tanh\left(\mathbf{W}\mathbf{s}_{n-1} + \mathbf{V}\mathbf{h}_t + \mathbf{U}\mathbf{f}_{nt} + \mathbf{b}\right), \tag{8}$$

where "$*$" means the 1-dimensional convolution operation, $\mathbf{F} \in \mathbb{R}^{C \times I}$ is a set of convolution kernels, $\mathbf{f}_n \in \mathbb{R}^{C \times T}$ is the result of the convolution, and C and I indicate the number of kernels and the size of each kernel. $\mathbf{w} \in \mathbb{R}^A$ is a weight vector, $\mathbf{W} \in \mathbb{R}^{A \times D}$, $\mathbf{V} \in \mathbb{R}^{A \times 2E}$, and $\mathbf{U} \in \mathbb{R}^{A \times C}$ are weight matrices, $\mathbf{b} \in \mathbb{R}^A$ is a bias vector, and A is the number of rows of $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{U}$, as well as the number of elements of $\mathbf{b}$.

Let $\mathbf{g}_n^{\mathrm{iso}} = [g_{n1}, \ldots, g_{nE}]^\top$ and $\mathbf{g}_n^{\mathrm{mix}} = [g_{n,E+1}, \ldots, g_{n,2E}]^\top$ denote the parts of $\mathbf{g}_n$ calculated from $\mathbf{h}^{\mathrm{iso}}$ and $\mathbf{h}^{\mathrm{mix}}$ in (2). The generation process of y_n in (3) is given by

$$\boldsymbol{\phi}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{p}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{p}}\mathbf{g}_n^{\mathrm{iso}} + \mathbf{b}^{\mathrm{p}}\right), \tag{9}$$

$$p_n = \underset{p \in V^{\mathrm{p}}}{\mathrm{argmax}} \left(\phi_p^{(n)}\right), \tag{10}$$

$$\boldsymbol{\psi}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{o}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{o}}\mathbf{g}_n^{\mathrm{iso}} + \mathbf{b}^{\mathrm{o}}\right), \tag{11}$$

$$o_n = \underset{o \in V^{\mathrm{o}}}{\mathrm{argmax}} \left(\psi_o^{(n)}\right), \tag{12}$$

$$\boldsymbol{\eta}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{b}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{b}}\mathbf{g}_n^{\mathrm{mix}} + \mathbf{b}^{\mathrm{b}}\right), \tag{13}$$

$$b_n = \underset{b \in V^{\mathrm{b}}}{\mathrm{argmax}} \left(\eta_b^{(n)}\right), \tag{14}$$

$$\boldsymbol{\xi}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{d}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{d}}\mathbf{g}_n^{\mathrm{mix}} + \mathbf{b}^{\mathrm{d}}\right), \tag{15}$$

$$d_n = \underset{d \in V^{\mathrm{d}}}{\mathrm{argmax}} \left(\xi_d^{(n)}\right), \tag{16}$$

where $\mathbf{P}^\star \in \mathbb{R}^{|V^\star| \times D}$, $\mathbf{Q}^\star \in \mathbb{R}^{|V^\star| \times E}$ are weight matrices, and $b^\star \in \mathbb{R}^{|V^\star|}$ is a bias parameter. Here, "$\star$" represents "p" (pitch), "o" (onset), "b" (beat), or "d" (downbeat). Note that $\mathbf{g}_n^{\mathrm{iso}}$ is used for estimating the pitches and onsets that are components of musical notes of a sung melody and $\mathbf{g}_n^{\mathrm{mix}}$ is used for estimating the beats and downbeats from the percussive sounds in the mixture sound.

(4) represents the recursive calculation of the next state $\mathbf{s}_n$. We adopt the teacher forcing for training. In short, the concatenation of one-hot vectors separately converted from the ground-truth pitch, onset, beat, and downbeat is used as y_n. The proposed model is optimized by minimizing the sum of the cross entropies for the elements of each y_n and the additional losses described in the next subsection. In the inference phase, the output symbols obtained by (10), (12), (14), and (16) at the previous step are converted into one-hot vectors and used for the concatenation of the one-hot vectors for predicting the current symbol. This recursive process is stopped when the output sequence reaches a predefined maximum length or when $\langle \mathtt{eos} \rangle$ symbol is generated as p_n, o_n, b_n, or d_n.

3.4. Loss functions for attention weights

We introduce new loss functions for the attention weights $\boldsymbol{\alpha}_{1:N} \in \mathbb{R}^{N \times T}$ to satisfy the *monotonicity* and *regularity* constraints mentioned in Section 1. In appropriate input-output alignment, the attention weights of each $\boldsymbol{\alpha}_n$ are known to be biased toward a narrow region in the time axis. As a representative point of each $\boldsymbol{\alpha}_n$, we use the centroid given by

$$c_n = \sum_{t=1}^{T} t \cdot \alpha_{nt}. \tag{17}$$

The loss function regarding monotonicity is given by

$$\mathcal{L}^{\mathrm{mono}} = \frac{1}{N-1} \sum_{n=1}^{N-1} \mathrm{ReLU}\left(-\Delta c_n\right), \tag{18}$$

where $\Delta c_n = c_{n+1} - c_n$ is the difference of the consecutive centroids, and ReLU is a rectified linear function given by $\mathrm{ReLU}(x) = \max(0, x)$. (18) prevents the order of the centroids from being reversed by imposing the positive cost only if the order of adjacent centroids is reversed.

The loss function regarding regularity is given by

$$\mathcal{L}^{\mathrm{reg}} = \frac{1}{N-2} \sum_{n=1}^{N-2} \left|\Delta c_{n+1} - \Delta c_n\right|^2. \tag{19}$$

This function makes the centroids be arranged at almost equal intervals that do not suddenly change over time.

4. EVALUATION

This section reports the results of comparative evaluations on the performance of the proposed method.

4.1. Data

To evaluate our model, we used 54 popular songs with reliable annotations from the RWC Music Database [32]. We split the input audio signals and the corresponding tatum-level scores into segments of 8 secs with an overlap of 1 sec. When we generated the tatum-level scores and split them, we referred to the annotated musical scores and beat times [33]. If the tatum crossed the start boundary of the segment, the tatum was removed from the segment. The tatum crossing the end boundary of the segment remained in the segment. We did not use segments including only rests.

All songs were sampled at 44.1 kHz, and we used a short-time Fourier transform (STFT) with a Hann window of 2048 points and a shifting interval of 441 points (10 ms) for calculating magnitude

Table 1: Error rates [%] in tatum and note levels.

Method	Attention loss		Tatum-level error rate					Note-level error rate [12]				
	Mono.	Reg.	$E_{\mathrm{p}}^{\mathrm{T}}$	$E_{\mathrm{o}}^{\mathrm{T}}$	$E_{\mathrm{b}}^{\mathrm{T}}$	$E_{\mathrm{d}}^{\mathrm{T}}$	$E_{\mathrm{a}}^{\mathrm{T}}$	$E_{\mathrm{p}}^{\mathrm{N}}$	$E_{\mathrm{m}}^{\mathrm{N}}$	$E_{\mathrm{e}}^{\mathrm{N}}$	$E_{\mathrm{on}}^{\mathrm{N}}$	$E_{\mathrm{off}}^{\mathrm{N}}$
Proposed	✓	✓	67.5	29.4	15.7	18.1	77.2	19.5	43.7	67.5	61.1	48.8
	✓		42.6	26.0	16.4	18.8	58.2	20.1	20.9	42.2	55.8	42.0
			34.6	22.5	15.4	17.8	48.3	18.3	11.4	31.7	46.3	34.1
Majority-vote	n/a	n/a	34.8	25.1	n/a	n/a	n/a	20.4	11.3	51.3	55.5	51.7

spectrograms, which were normalized to make the maximum value equal to 1, followed by the calculation of the mel-scale spectrograms with 229 channels. The mel-scale spectrograms of training data were standardized for each frequency bin, and the mel-scale specgtrograms of evaluation data were standardized for each frequency bin by using means and standard deviations calculated from the training data.

4.2. Setup

The vocabulary of pitches consisted of rest, 40 semitone-level pitches from E2 to G5 ($K = 41$), and the special elements $\langle \mathtt{sos} \rangle$ and $\langle \mathtt{eos} \rangle$. The pitch vocabulary covered pitches contained in both the train and test data. We assumed that an input sung melody can be represented as a monophonic sequence of musical notes.

Each encoder consisted of a three-layer bidirectional LSTM with 100×2 cells with the dropout rate of 0.2. The decoder consisted of one-layer LSTM with 100 cells. A padding size and a stride of convolution in the attention mechanism were set to 50 and 1, respectively, and the parameters were $C = 10$, $I = A = 100$. Adam [34] was used to optimize the parameters of the proposed model, and a weight decay (L2 regularization) with a controllable hyperparameter 10^{-5} and a gradient clipping with a threshold of 5.0 were applied for training. All weight parameters of fully-connected layers were initialized with random values drawn from the uniform distribution $\mathcal{U}(-0.1, -0.1)$, and all bias parameters were initialized with zeros. The kernels of the CNN in the attention mechanism and the weight parameters of the encoder and decoder were initialized by He's method [35]. The maximum lengths in the inference phase was set to 200. The batch size and the number of epochs were 150 and 100, respectively. PyTorch v1.0.1 was used for implementation.

4.3. Metrics

The performance of tatum-level transcription was measured using tatum-level error rate defined as:

$$\frac{N_S + N_D + N_I}{N} \times 100\ [\%], \tag{20}$$

where the numerator represents the Levenshtein distance between the ground-truth and prediced scores: N_S, N_D, and N_I are the minimum number of substitutions, deletions, and insertions required to change the predicted sequence into the ground-truth N is the number of tatums in the ground-truth. We used the error rate for evaluating each of a pitch $E_{\mathrm{p}}^{\mathrm{T}}$, an onset flag $E_{\mathrm{o}}^{\mathrm{T}}$, a beat flag $E_{\mathrm{b}}^{\mathrm{T}}$, and a downbeat flag $E_{\mathrm{d}}^{\mathrm{T}}$ and all of them $E_{\mathrm{a}}^{\mathrm{T}}$.

To measure the note-level performance, we used the metrics proposed in [12] that calculate the following five values: pitch error rate $E_{\mathrm{p}}^{\mathrm{N}}$, missing note error rate $E_{\mathrm{m}}^{\mathrm{N}}$, extra note rate $E_{\mathrm{e}}^{\mathrm{N}}$, onset-time error rate $E_{\mathrm{on}}^{\mathrm{N}}$, and offset-time error rate $E_{\mathrm{off}}^{\mathrm{N}}$. In constructing of the score from the predicted symbols, we applied the following rules:

1. If $p_{n-1} \neq p_n$, then the $(n-1)$-th and n-th tatums are included in different notes.
2. If $p_{n-1} = p_n$ and $o_n = 1$, then the $(n-1)$-th and n-th tatums are included in different notes having the same pitch.
3. If $p_{n-1} = p_n$ and $o_n = 0$, then the $(n-1)$-th and n-th tatums are included in the same note.

Note that the note-level metrics do not take into account rests, beats and downbeats. We used the parameters that minimize an epoch average of E_p^T calculated using some of the evaluation data.

4.4. Results

Experimental results in Table 1 showed that both tatum- and note-level error rates became worse by using the proposed loss functions for attention weights. The reason for this result is that the constraint of the loss functions is too strong to prevent the attention mechanism from finding appropriate position in the input sequence. In the early stages of training, the centroids usually line up at the roughly same positions, and $\mathcal{L}^{\mathrm{mono}}$ and $\mathcal{L}^{\mathrm{reg}}$ are almost zero. The centroids cannot escape from the initial positions because the values of $\mathcal{L}^{\mathrm{mono}}$ and $\mathcal{L}^{\mathrm{reg}}$ increase when the centroids change.

We also compared the proposed method to the majority-vote method used in [25] using ground-truth F0 trajectory with voice activity detection and tatum times. A boundary of pitch changes was regarded as note onset positions and the successive tatums having the same pitch were included in one note. Since the method does not estimate beats and downbeats, $E_{\mathrm{b}}^{\mathrm{T}}$, $E_{\mathrm{d}}^{\mathrm{T}}$, and $E_{\mathrm{a}}^{\mathrm{T}}$ are not used. In most metrics, the proposed method without the attention losses attained better error rates than the majority-vote method. Especially, we obtained an improvement on E_e^N.

5. CONCLUSION

This paper presented the method for musical note transcription of a sung melody based on the encoder-decoder model with the beat-synchronous attention mechanism. We extended the standard encoder-decoder model to simultaneously predict a pitch, an onset flag, a beat flag, and a downbeat flag that are needed to construct a musical score. We also proposed the loss functions to induce the attention weights to have proper structure. We experimentally investigated whether those loss functions are effective or not.

For future work, we can incorporate other tasks of music analysis in the proposed model. We can integrate a model for separating isolated singing voice from an audio mixture, which are separately fed to the proposed model. Owing to the tatum-level representation of output symbols, other symbols (*e.g.*, chords) in musical score can be easily added into the output of the proposed method. These integrations would improve the transcription accuracy and facilitate the simultaneous analysis of music.

6. REFERENCES

[1] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *ICASSP*, 2012, pp. 57–60.

[2] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *ISMIR*, 2017, pp. 23–27.

[3] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *ICASSP*, 2014, pp. 659–663.

[4] Y. Ikemiya, K. Yoshii, and K. Itoyama, "Singing voice analysis and editing based on mutually dependent F0 estimation and source separation," in *ICASSP*, 2015, pp. 574–578.

[5] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music." in *ISMIR*, 2017, pp. 63–70.

[6] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *ICASSP*, 2018, pp. 161–165.

[7] E. Molina, L. J. Tardón, A. M. Barbancho, and I. Barbancho, "SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 2, pp. 252–263, 2015.

[8] N. Kroher and E. Gómez, "Automatic transcription of flamenco singing from polyphonic music recordings," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 5, pp. 901–913, 2016.

[9] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," in *TENOR*, 2015, pp. 23–30.

[10] L. Yang, A. Maezawa, J. B. L. Smith, and E. Chew, "Probabilistic transcription of sung melody using a pitch dynamic model," in *ICASSP*, 2017, pp. 301–305.

[11] H. Takeda, N. Saito, T. Otsuki, M. Nakai, H. Shimodaira, and S. Sagayama, "Hidden markov model for automatic transcription of MIDI signals," in *MMSP*, 2002, pp. 428–431.

[12] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 794–806, 2017.

[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015, pp. 1412–1421.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015, pp. 1–15.

[15] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015, pp. 577–585.

[16] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016, pp. 4960–4964.

[17] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of "attention" in sequence-to-sequence models," in *INTERSPEECH*, 2017, pp. 3702–3706.

[18] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *ICASSP*, 2019, pp. 161–165.

[19] C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *ICLR*, 2018.

[20] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *ICML*, 2017, pp. 2837–2846.

[21] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1727–1741, 2013.

[22] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *ISMIR*, 2018, pp. 50–57.

[23] C. Raphael, "A hybrid graphical model for rhythmic parsing," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 217–238, 2002.

[24] M. Hamanaka, M. Goto, H. Asoh, and N. Otsu, "A learning-based quantization: Unsupervised estimation of the model parameters," in *ICMI*, 2003, pp. 369–372.

[25] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Scale- and rhythm-aware musical note estimation for vocal F0 trajectories based on a semi-tatum-synchronous hierarchical hidden semi-markov model," in *ISMIR*, 2017, pp. 376–382.

[26] E. Nakamura, R. Nishikimi, S. Dixon, and K. Yoshii, "Probabilistic sequential patterns for singing transcription," in *APSIPA ASC*, 2018, pp. 1905–1912.

[27] R. G. C. Carvalho and P. Smaragdis, "Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score," in *WASPAA*, 2017, pp. 151–155.

[28] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "An end-to-end framework for audio-to-score music transcription on monophonic excerpts," in *ISMIR*, 2018, pp. 34–41.

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104–3112.

[30] H.-W. Nienhuys and J. Nieuwenhuizen, "Lilypond, a system for automated music engraving," in *CIM*, 2003, pp. 167–171.

[31] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[32] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, classical and jazz music databases," in *ISMIR*, 2002, pp. 287–288.

[33] M. Goto, "AIST annotation for the RWC music database." in *ISMIR*, 2006, pp. 359–360.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, 2014, pp. 1–15.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.