CT-SAT: Contextual Transformer for Sequential Audio Tagging

Yuanbo Hou∗1, Zhaoyi Liu2, Bo Kang1, Yun Wang3, Dick Botteldooren∗1

∗WAVES Research Group 1Ghent University, Belgium 2KU Leuven, Belgium 3Meta AI, USA

{Yuanbo.Hou, Bo.Kang, Dick.Botteldooren}@UGent.be
Zhaoyi.Liu@student.kuleuven.be, Maigoakisame@gmail.com

Abstract

Sequential audio event tagging can provide not only the type information of audio events, but also the order information between events and the number of events that occur in an audio clip. Most previous works on audio event sequence analysis rely on connectionist temporal classification (CTC). However, CTC’s conditional independence assumption prevents it from effectively learning correlations between diverse audio events. This paper first introduces the Transformer into sequential audio tagging, since Transformers perform well in sequence-related tasks. To better utilize contextual information of audio event sequences, we draw on the idea of bidirectional recurrent neural networks, and propose a contextual Transformer (cTransformer) with a bidirectional decoder that could exploit the forward and backward information of event sequences. Experiments on the real-life polyphonic audio dataset show that, compared to CTC-based methods, the cTransformer can effectively combine the fine-grained acoustic representations from the encoder and coarse-grained audio event cues to exploit contextual information to successfully recognize and predict the audio event sequence in polyphonic audio clips.

Index Terms: Audio tagging, sequential audio tagging, connectionist temporal classification, contextual Transformer

1. Introduction

Audio Tagging (AT) is a multi-label classification task that identifies which target audio events occur in an audio clip. AT only predicts the type of events occurring in an audio clip, not the order between these events nor how many times they occur. As audio events naturally occur sequentially in a sequence, there is often a relationship between the preceding and following events. This paper studies sequential audio tagging (SAT), which aims to learn such relationships between events and predict sequences of audio events in audio clips. SAT can be applied for tasks such as audio classification [1], audio captioning [2], acoustic scene analysis [3], and event anticipation [4].

Previous works related to SAT mostly rely on connectionist temporal classification (CTC) [5] to identify event sequences. Paper [6] explores the possibility of polyphonic SAT using sequential labels and utilizes CTC to train convolutional recurrent neural networks (CRNN) [7] with learnable gated linear units (GLU) [8] to tag event sequences. As audio events often overlap with each other, the order of start and end boundaries of events are used in [6] as sequential labels. For example, the double-boundary sequential label of an audio clip might be “dishes_start, dishes_end, speech_start, blender_start, speech_end, speech_start, blender_end, speech_end”. Sequential labels do not contain the onset and offset time information of audio events, which avoids the problem of inaccurate annotations of frame-level labels, and reduces the annotation workload. In addition to exploring the feasibility of recognizing audio event sequences in SAT, CTC-based methods have also been proposed for sound event detection (SED), which detects the type, starting time, and ending time of audio events. A bidirectional long short-term memory (LSTM) RNN [9] equipped with CTC (BLSTM-CTC) [10] is used to detect events using double-boundary sequential labels. The results [10] on a very noisy corpus show that BLSTM-CTC is able to locate boundaries of audio events with rough hints about their time positions. Apart from methods using double-boundary labels, another CTC-based SES system [11] uses single-boundary sequential labels (the sequence of start boundaries of events) with unsupervised clustering to detect the type and occurrence time of audio events. CTC defines the loss function of RNN [5] and allows it to be trained for sequence tasks to keep order information of events in the sequence. However, CTC implicitly assumes that outputs of the network at different time steps are conditionally independent [5], which makes CTC-based approaches unable to effectively learn the contextual information inherent in audio event sequences. This paper introduces the Transformer [12], which has revolutionized the field of natural language processing [13], into SAT. The Transformer [12] does not have the conditional independence assumption in CTC. Compared with RNN-based models, the Transformer can access information at any time step from any other time step, thereby capturing long-term dependencies [14] between audio events. In addition, the training of Transformers can also be efficiently parallelized.

When learning sequence information, the decoder in the Transformer exploits past information to infer the upcoming event. For example, when recognizing audio event sequences “fire, alarm, run” and “fire, crying, sobbing”, the model may be confused between alarm and crying when forward inferring the next event from fire. But if the target event is backward inferred from run and sobbing respectively, the probability of alarm and crying is different in different sequences. Contextual information can help the model learn the differences between sequences in detail. To make more comprehensive use of the contextual information in audio event sequences, this paper draws on the idea of bidirectional RNN [15] and proposes a contextual Transformer (cTransformer) to explore bidirectional information in audio event sequences. The cTransformer consists of an encoder and a decoder, the latter of which is the main focus of this paper. The decoder attempts to fuse frame-level representations from the encoder with the clip-level event cues to infer the target by combining the forward and backward information learned from normal and reverse sequences, respectively. Then, the loss between the prediction from the normal sequence branch and the prediction from the reverse sequence branch is calculated and back-propagated to update parameters to learn a more consistent prediction about the same target. In training, partial weights of the normal and reverse sequence branches are shared to learn forward and backward information. With the help of these shared weights, the decoder is able to learn contextual information from both directions simultaneously to more comprehensively and accurately identify event sequences.
The contributions of this paper are: 1) we introduce the Transformer into SAT; 2) We propose the cTransformer that can utilize bidirectional information to better identify audio event sequences in audio clips; 3) To explore the feasibility of SAT based on cTransformer, we manually label sequential labels for a polyphonic audio dataset from real life, and compare the performance of the cTransformer and other CTC-based methods on it. This paper is organized as follows, Section 2 introduces the cTransformer. Section 3 describes the dataset, experimental setup, and analyzes the results. Section 4 gives conclusions.

2. Contextual Transformer

Motivated by the performance of Transformers in sequence tasks [12, 16] and the significance of contextual information in audio tasks [17, 18, 19], this paper proposes the cTransformers for audio event sequence analysis. The cTransformer aims to transform an audio clip to the corresponding sequence of event labels using both global information and rich contextual details.

2.1. Input and output definition

The most common acoustic feature for acoustic event recognition is the log mel spectrogram [20]. We convert every audio clip $x$ into its log mel spectrogram $X(t, f)$ as the model input. Following [11], the sequence of event start boundaries is used as sequential labels. For the normal sequence branch in Figure 1, the label sequence $y$ is "$<$S$>$, event1, event2, ..., event$k$, $<$E$>$", where $k$ is the number of event occurrences, and $<$S$>$ and $<$E$>$ are tokens indicating the start and end of prediction, respectively. For the reverse sequence branch, the label sequence $y'$ is "$<$S$'$, event1, event2, ..., event$k$, $<$E$'$>", where $<$S$'$> is the token indicating the start of reverse sequence prediction. Note that we use different start tokens, but the same end token for the two directions. The sequential label $y$ of an audio clip may be "$<$S$>$, dishes, speech, speech, blender, speech, $<$E$>$"; the corresponding $y'$ is "$<$S$'$>, speech, blender, speech, speech, dishes, $<$E$'$>".

2.2. The encoder part of the contextual Transformer

The encoder aims to convert input acoustic features into high-level representations. To consider the audio information globally, this paper does not divide input features into small patches [21], so there is no positional encoding [12] in the encoder. The encoder mainly consists of $N$ identical blocks with multi-head attention layers (MHA) and feed forward layers, which are analogous to the encoder in the Transformer [12]. The attention function in MHA is scaled dot-product attention, whose input consists of queries and keys of dimension $d_k$ and values of dimension $d_v$ [12]. The attention is calculated on a set of queries, keys, and values packed into matrix $Q$, $K$, and $V$, respectively.

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) V$$

(1)

Then, MHA is used to allow the model to jointly focus on representations from different subspaces at different positions.

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$

(2)

where \(\text{head}_i = \text{Attention}(QW^Q_i, KW^K_i, WV^V_i)\)

The decoder consists of two branches with the same structure. Each branch contains $M$ identical blocks in series, each of which contains a masked MHA layer, an MHA layer, and a feed-forward layer, analogous to the decoder in the original Transformer [12]. The masked MHA layer uses self-attention, i.e., $Q$, $K$, and $V$ all come from the input sequence of event labels. To preserve the autoregressive property, forward and backward masks are applied to block future and past information, respectively; masked positions in the attention map $QK^T$ are replaced with $-\infty$ before the softmax operation. Next, an MHA layer is used to fuse frame-level acoustic representations from the encoder with clip-level event cues from the previous decoder layer. In this layer, $Q$ comes from the previous masked MHA layer, while $K$ and $V$ come from the output of the encoder. When predicting the $t$-th target $\text{event}_t$, given the input cue for the normal sequence branch is $\overline{Z}_{t-1}$ and the input cue for the reverse sequence branch is $\overline{Z}_{t+1}$. Let $\overline{p}_{t}$ and $\overline{p}_{t}$ be...
the prediction for event\textsubscript{t} from the normal and reverse sequence branches. For the normal sequence branch exploring forward information, $\overline{p}_{t}$ is derived from the output of encoder $O_{En}$ and $\overline{x}_{t-1}$, after forward Masked MHA $M_{f}$. For the reverse sequence branch exploring backward information, $\overline{p}_{t}$ is derived from $O_{En}$ and $\overline{x}_{t+1}$ after backward Masked MHA $M_{b}$.

$$\overline{p}_{t} = \phi(M_{f}(\overline{x}_{t-1}) + O_{En}W_{MHA}^{(f)} + b^{(f)})$$

$$\overline{p}_{t} = \phi(M_{b}(\overline{x}_{t+1}) + O_{En}W_{MHA}^{(b)} + b^{(b)})$$

where $b^{(f)}$ and $b^{(b)}$ are biases in normal and reverse sequence branches, $W_{MHA}^{(f)}$ and $W_{MHA}^{(b)}$ are learnable weights in MHA, $\phi$ denote the set of mapping functions in each branch of the decoder. The remaining layers and parameters in the decoder are the same as those of the Transformer [12].

### 2.4. The loss function of the contextual Transformer

The loss function is a summation of the loss over each step in the event label sequences of the training data. Denote $\overline{p}_{t}$ and $\overline{x}_{t}$ as $p_{t}$ and $y_{t}$. At each step, $p_{t}$ and $y_{t}$ are the distribution of labels predicted by the normal and reverse sequence branches, while the ground-truth labels are $y_{t}$ and $y_{t}$. Following the loss function in the original Transformer [12], the cross entropy (CE) loss is computed for both branches:

$$L_{\text{normal}} = CE(p_{t}, y_{t}) \quad L_{\text{event}} = CE(p_{t}, y_{t})$$

(4)

Since $p_{t}$ and $p_{t}^{'}$ are the prediction for the same target, the mean squared error (MSE) loss that performs well in regression tasks [23][24][25] is used as the context loss to measure the distance between $p_{t}$ and $p_{t}^{'}$. In the latent space.

$$L_{\text{context}} = MSE(p_{t}, p_{t}^{'}$$)

(5)

To consider both forward and backward information at the same time in the training, the different types of losses of different branches are added together. The final loss of the model is

$$L = \lambda_{l}L_{\text{normal}} + \lambda_{c}L_{\text{event}} + \lambda_{c}L_{\text{context}}$$

(6)

where $\lambda$ adjusts the weights of loss components during training. Each $\lambda$ defaults to 1. During the training process, the forward prediction $p_{t}$ and backward prediction $p_{t}^{'}$ will be aligned to capture the rich contextual information around the target event and learn the entire sequence embeddings more accurately.

### 3. Experiments and results

#### 3.1. Dataset, Baseline, Experiments Setup, and Metrics

Since there is no publicly available polyphonic audio dataset with sequential labels, we manually label the DCASE domestic environment audio dataset [26] with the sequences of event start boundaries as sequential labels following [11]. The audio dataset excerpted from Audiodot [27] contains 10 classes of real-life audio events, where the training and test sets consist of 1578 and 288 audio clips, respectively. The number of event occurrences contained in the training and test sets is 3619 and 923. During training, 20% of the training samples are randomly selected and set aside as the validation set.

Since most previous works on audio event sequence analysis rely on CTC, the BLSTM-CTC [10] is used as Baseline.

<table>
<thead>
<tr>
<th>#</th>
<th>$[N, M]$</th>
<th>AUC</th>
<th>BLEU</th>
<th>#</th>
<th>$[N, M]$</th>
<th>AUC</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 1</td>
<td>0.771</td>
<td>0.468</td>
<td>7</td>
<td>3, 3</td>
<td>0.784</td>
<td>0.482</td>
</tr>
<tr>
<td>2</td>
<td>1, 2</td>
<td>0.800</td>
<td>0.491</td>
<td>8</td>
<td>3, 6</td>
<td>0.770</td>
<td>0.472</td>
</tr>
<tr>
<td>3</td>
<td>2, 2</td>
<td>0.775</td>
<td>0.481</td>
<td>9</td>
<td>4, 2</td>
<td>0.779</td>
<td>0.467</td>
</tr>
<tr>
<td>4</td>
<td>2, 4</td>
<td>0.775</td>
<td>0.483</td>
<td>10</td>
<td>4, 4</td>
<td>0.787</td>
<td>0.464</td>
</tr>
<tr>
<td>5</td>
<td>2, 5</td>
<td>0.783</td>
<td>0.473</td>
<td>11</td>
<td>5, 5</td>
<td>0.774</td>
<td>0.461</td>
</tr>
<tr>
<td>6</td>
<td>3, 1</td>
<td>0.782</td>
<td>0.474</td>
<td>12</td>
<td>6, 6</td>
<td>0.778</td>
<td>0.456</td>
</tr>
</tbody>
</table>

#### 3.2. Results and Analysis

##### Number of encoder and decoder blocks

The encoder and decoder of the cTransformer consist of $N$ and $M$ identical blocks, respectively. This paper first explores the optimal ratio of blocks of encoder and decoder to determine the final model structure, as shown in Table 1. We choose AUC to represent the models’ AT performance, since AUC does not depend on the threshold.

To evaluate the performance of the model does not increase monotonically with the number of blocks. The best results on the test set are achieved with $[N, M] = [1, 2]$, which are significantly smaller than the default setting $[N, M] = [6, 6]$ of the original Transformer [12]. The reason why our best model is smaller may be that the polyphonic audio dataset we have manually labeled is not large enough. In experiments, we have observed serious overfitting as the values of $N$ and $M$ get large. So $[N, M] = [1, 2]$ will be used in the following experiments.

##### Weighting factors in the loss

The next step is to optimize the weighting factors $\lambda$ of different loss terms. The $L_{\text{normal}}$ and $L_{\text{context}}$ focus on learning task-goal-oriented representations to improve the accuracy of individual event sequence recognition, while the $L_{\text{context}}$ aims to align the predictions of the normal and reverse sequence branches to make them more consistent.

Table 2 presents an ablation study to demonstrate the necessity of each term in cTransformer’s loss function. Model #1 has only the normal sequence branch, and is equivalent to the original Transformer [12]. Conversely, Model #2 has only the reverse sequence branch. Except for the result of # 2 from the reverse sequence branch, the rest are predicted by the normal sequence branch. Model #3, which has both the normal and the reverse sequence branches, outperforms #1 and #2 which only

<table>
<thead>
<tr>
<th>#</th>
<th>$L_{\text{normal}}$</th>
<th>$L_{\text{context}}$</th>
<th>$L_{\text{context}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>3</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>4</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
</tbody>
</table>

in this paper. This paper also compares the cTransformer with CTC-based convolutional bidirectional gated recurrent units (CBGRU-CTC) [28], and CBGRU-CTC equipped with GLU in convolutional layers (CBGRU-GLU-CTC) [6], and in both convolutional and recurrent layers (CBGRU-GLU-CTC) [11].

As the input feature, log mel-band energy with 64 banks [20] is extracted using STFT with Hamming window length of 46 ms and the overlap is 1/3 between windows, following the settings of [29]. During training, stochastic gradient descent with momentum (SGDM) [30] with an initial learning rate of 1e-3, a batch size of 64, and a momentum value of 0.9 is used to minimize the loss. Dropout [31] and layer normalization [32] are used to prevent overfitting. Systems are trained on a single card Tesla V100-SXM2-32GB for a maximum of 1000 epochs. For more details and the manually labeled dataset with sequential labels, please visit the project homepage (https://github.com/Yuanbo2020/Contextual-Transformer).

The output of SAT consists of the types of audio events plus the order information between them. This paper uses precision (P), recall (R), F-score (F), accuracy (Acc) [33], and area under curve (AUC) [34] to measure various aspects of the models’ performance in recognizing the types of audio events, and adopts the bilingual evaluation understudy (BLEU) [35] metric commonly used in sequence tasks to evaluate the models’ ability to recognize the order between events. Higher P, R, F, Acc, AUC, and BLEU indicate better performance.
Figure 2: Attention score from the masked MHA in decoder. Subgraph (a) and (b) are from the normal and reverse sequence branches, respectively. The x-axis is each event predicted in an autoregressive way, y-axis is the corresponding reference event.

have a single branch. Model #4, which is further equipped with the context loss, performs the best of all. By encouraging both branches to yield consistent predictions, the model is better able to integrate the information contained in both directions to more accurately identify and effectively confirm the target event.

Table 3 further tunes the weights of the different loss terms in a fine-grained manner to find out the optimal combination of coefficients. Table 3 attempts to control variables to compare the performance of models with different combinations of coefficients. Finally, giving the same weight to $L_{\text{normal}}$ and $L_{\text{context}}$, and lightening the weight of $L_{\text{reverse}}$ achieves the best AUC in # 4. This reveals that in the experiments, the cTransformer should focus on capturing the forward and contextual information, while putting the backward information in a secondary position for better event sequence recognition.

Comparison with prior methods. In Table 4, the cTransformer is compared with prior methods for recognizing audio event sequences. All models are retrained on the DCASE dataset with manually labeled event sequences. Since previous works using CTC-based methods did not apply data augmentation, we do not apply data augmentation for a fair comparison. To analyze the ability of different models to identify polyphonic audio events from multiple perspectives, multiple metrics are adopted for AT and the BLEU metric is used for SAT.

As shown in Table 4, BLSTM-CTC [10], which uses only LSTM to extract acoustic representations for recognizing polyphonic audio event sequences, has the worst overall performance. The CBGRU-CTC [28] with a composite convolutional recurrent neural network outperforms the BLSTM-CTC on most metrics, which demonstrates the superior ability of convolutional layers in feature extraction. CGLU-BGRU-CTC [6] and CBGRU-Glu-CTC [11] with GLU assembled in convolutional layers and both convolutional and recurrent layers, respectively, do not bring further overall improvement, although they do outperform CBGRU-CTC in some metrics. Table 4 also shows the results of the original Transformer [12] with a 6-layer encoder and decoder. Possibly due to the limited size of the polyphonic audio dataset, the performance of Transformer is close to that of the CTC-based methods. Overall, the cTransformer achieves the best results in both AT and SAT.

Case study. To gain a more intuitive insight into the performance of the model on polyphonic audio event sequences, we conduct a case study on an audio clip where the ground-truth sequence of event start boundaries is “frying, dishes, dishes”. Figure 2 shows the distribution of attention scores from the masked MHA of the normal and reserve sequence branches. In Figure 2 (a), after inputting $<$S$, the attention value for $<$S$>$ is 1, then combining acoustic representations from the encoder, the model predicts the next event should be $<$frying, the event corresponding to the 2nd column of the x-axis). The reference event label is $frying$ (the event corresponding to the 1st row of the y-axis). Then, the input is $<$S$, $frying$, attention values for the two events are 0.34 and 0.66, respectively. The next event is predicted to be $<$dishes$>$ (the event corresponding to the 3rd column of the x-axis). The reference label is $dishes$ (the event corresponding to the 2nd row of the y-axis). Finally, when the input is $<$S$>, $frying, dishes, dishes$>, based on acoustic representations, the model judges that the event sequence is complete, and subsequently outputs $<$E$>$ (the event corresponding to the 4th row of the y-axis) to indicate the inference stops. After the autoregressive process, the predicted event sequence $p$ = “frying, dishes, dishes” is obtained, the reference label sequence $y$ is “dishes, dishes, frying”. The match between $p$ and $y$ indicates that the cTransformer successfully fuses frame-level acoustic representations from the encoder with clip-level event cues from the decoder to jointly infer the event sequence. In Figure 2 (b), attention scores from reverse sequence branch for the same audio clip are different from attention scores for forward inference in Figure 2 (a). Guided by $<$S$>$, the reverse sequence branch combining audio representations successfully predicts the reverse sequence $p^r$ = “dishes, dishes, frying”, the corresponding label $y^r$ is “dishes, dishes, frying”. The match of $p^r$ and $y^r$ indicates that with the assistance of different prediction cues and mask matrices, the cTransformer effectively infers the event sequence from normal and reverse directions, which implies that the model is effective for modeling contextual information.

4. Conclusions
This paper first introduces the Transformer into SAT. To utilize the context information of audio event sequences, this paper proposes cTransformer with a bidirectional sequence decoder, which can exploit both forward and backward information. The cTransformer automatically assigns different attention scores to available information to effectively model contextual information and infer the event, and then efficiently fuses frame-level acoustic representations and clip-level event cues to identify event sequences implicit in audio clips. Future work will explore the performance of cTransformer using fully bidirectional information to infer audio event sequences on more datasets.

5. ACKNOWLEDGEMENTS
The WAVES Research Group and Bo Kang received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

<table>
<thead>
<tr>
<th>Method</th>
<th>AT P(%)</th>
<th>AT K(%)</th>
<th>SAT P(%)</th>
<th>SAT K(%)</th>
<th>1 AUC BLEU</th>
<th>2 AUC BLEU</th>
<th>4 AUC BLEU</th>
<th>6 AUC BLEU</th>
<th>8 AUC BLEU</th>
<th>10 AUC BLEU</th>
<th>12 AUC BLEU</th>
<th>14 AUC BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLSTM-CTC [10]</td>
<td>69.73</td>
<td>30.12</td>
<td>58.32</td>
<td>89.47</td>
<td>0.713</td>
<td>0.323</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBGRU-CTC [28]</td>
<td>67.79</td>
<td>63.39</td>
<td>63.23</td>
<td>90.93</td>
<td>0.793</td>
<td>0.475</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGLU-BGRU-CTC [6]</td>
<td>77.87</td>
<td>60.99</td>
<td>69.17</td>
<td>90.48</td>
<td>0.786</td>
<td>0.468</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBGRU-Glu-CTC [11]</td>
<td>75.97</td>
<td>64.30</td>
<td>68.65</td>
<td>91.77</td>
<td>0.787</td>
<td>0.463</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformer [12]</td>
<td>67.24</td>
<td>64.53</td>
<td>65.86</td>
<td>90.17</td>
<td>0.785</td>
<td>0.432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cTransformer</td>
<td>75.66</td>
<td>67.61</td>
<td>71.41</td>
<td>92.05</td>
<td>0.805</td>
<td>0.505</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. References


