Abstract

Transformer-based models have demonstrated their effectiveness in automatic speech recognition (ASR) tasks and even shown superior performance over the conventional hybrid framework. The main idea of Transformers is to capture the long-range global context within an utterance by self-attention layers. However, for scenarios like conversational speech, such utterance-level modeling will neglect contextual dependencies that span across utterances. In this paper, we propose to explicitly model the inter-sentential information in a Transformer based end-to-end architecture for conversational speech recognition. Specifically, for the encoder network, we capture the contexts of previous speech and incorporate such historic information into current input by a context-aware residual attention mechanism. For the decoder, the prediction of current utterance is also conditioned on the historic linguistic information through a conditional decoder framework. We show the effectiveness of our proposed method on several open-source dialogue corpora and the proposed method consistently improved the performance from the utterance-level Transformer-based ASR models.

Index Terms: End-to-end speech recognition, Transformer, Long context, Conversational ASR

1. Introduction

Context information plays an important role in ASR, especially in scenes that require inter-sentential information such as conversation since semantically related words, or phrases often reoccur across sentences [1]. Typically, traditional hybrid acoustic-language ASR models usually rely on rich language models to model contextual information [2, 3, 4, 5, 6, 7]. Meanwhile, there are also several researches adopting context information particularly in end-to-end ASR by adding additional context to the decoder or simply concatenate multiple consecutive utterances as the input of an end-to-end model [8, 9, 10].

Transformer [11], as the most successful attention-based end-to-end model, has recently received more attention due to its superior performance on a wide range of tasks including ASR [12, 13, 14, 15, 16, 17]. However, since the computational and memory cost of self-attention is quadratic w.r.t the input sequence length, Transformer is hard to process long sequences and mainly models independent utterances.

Several studies in natural language processing (NLP) have been explored to utilize the long contextual information for Transformer [18, 19, 20, 21]. Inspired by above studies in the NLP task, some approaches were also proposed to incorporate contextual information across successive input sequences in Transformer-based ASR [10, 22], but these methods do not solve the problem of the high computational and memory cost, or have high model complexity.

In this study, we propose a novel Transformer-based architecture to explicitly model the inter-sentential information for conversational ASR. Inspired by [23], we include a residual attention module in the encoder, which accelerates the convergence speed and well models the long-range global dependencies within each input sequence. Besides, to further transfer the contextual information of previous sentences, we also propose a novel context-aware residual attention module, which transfers contextual information through attention scores. For the decoder part, we use an additional context module to learn more inter-sentential information. By using the methods above, we introduce inter-sentential contextual information in the popular Transformer ASR model. We demonstrate the superiority of our approach on two dialogue benchmarks (speech from two speakers) HKUST and Switchboard, a lecture benchmark TED-LIUM2, and a dialog dataset DATATANG-dialog, with obvious error rate reduction and neglectable increase of computational cost and model complexity.

2. Transformer and Conformer

Transformer [11] is an attention-based end-to-end model, which consists of multi-block stacked encoder and decoder. Each block can be characterized by a multi-headed attention (MHA) module, a position-wise feed-forward (FFN) module, layer normalization (LN) layers, and residual connections.

The MHA module calculates the score through the vectors $V^h$ and keys $K^h$, and assigns values to the output embeddings queries $Q^h$ [23]:

$$MHA(Q^h, K^h, V^h) = \text{Concat}(\text{heads}_1, \ldots, \text{heads}_m)W^O,$$ (1)

where $\text{heads}_i = \text{Attn}(QW^Q_i, KW^K_i, VW^V_i)$. Here, $Q$, $K$ and $V$ are matrices with dimension $d_k, d_k$ and $d_v$. $W^Q$, $W^K$ and $W^V$ are matrices that maps three vectors to the $i$-th head in the multi-head attention space. $W^O$ is a linear layer to transform the output after the stitching.

In the attention module, we use the traditional scaled dot-product attention module [11] to calculate the attention scores, as shown below:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$ (2)

where $\frac{QK^T}{\sqrt{d_k}}$ is a matrix representing the attention scores for each query and key.
The FFN module is composed of two fully-connected layers with a ReLU activation in between, as follows:

\[ \text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \]  

(3)

In addition to the two sub-layers in each encoder block, the decoder inserts a third sub-layer, which performs multi-headed attention over the source and target sequences.

Recently, Gulati et al. [17] combined Transformer and Convolutional Neural Networks (CNN) as Conformer. The Conformer encoder adds a convolution module between MHA and FFN in Transformer encoder blocks, simultaneously capturing local and global contextual information and leading to superior performance in ASR tasks.

3. Proposed Method

3.1. Residual Multi-headed Attention

Residual multi-headed attention (ResMHA) closely follows the same Post LN strategy as in [11], which normalizes the output at the end of each MHA or FFN module [23]. ResMHA connects MHA of adjacent layers through attention scores, as shown in Fig. 1 (a). Formally, it uses the attention score \( \text{Prev} \) of the previous layer as a conditional input to calculating the attention score of the current layer’s MHA. In particular, \( \text{Prev} \) is the attention score before the Softmax operation:

\[ \text{ResMHA}(Q^h, K^h, V^h, \text{Prev}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_m)W^Q, \]  

(4)

where \( \text{head}_i = \text{ResAttn}(QW^Q_i, KW^K_i, VW^V_i, \text{Prev}_i) \). Like \( \text{head}_i \), \( \text{Prev} \) is the slice of \( \text{Prev} \). Then these residual attention scores, which corresponding to the MHA heads, will be added to the current attention calculation through the Residual attention (ResAttn) module:

\[ \text{ResAttn}(Q, K, V, \text{Prev}) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}} + \text{Prev}_v)V. \]  

(5)

We apply it to speech Transformer, accelerating the convergence speed of the model during training and improving the final speech recognition accuracy.

3.2. Context-aware Multi-headed Residual Attention

Context-aware multi-headed residual attention is designed on the basis of Residual attention [23], which adds a skip edge to connect multi-headed attention (MHA) modules adjacent layers, as shown in Fig. 1 (a). To capture contextual information across different consecutive utterances, an intuitive idea is to simply concatenate previous inputs with the current input to the encoder. Although, such method can give a slight performance improvement, it will also confront a large increment of memory cost and computation complexity. Thus, we propose a context-aware multi-headed attention, which transfers the attention hidden states of the previous sentence in time order (StateLS) to the current sentence and straightforwardly includes more contextual information during the training. Fig. 1 (b) shows the details of our method. When we recognize the \( m \)-th sentence \( X_m \) in the conversation and use the previous sentence \( X_{m-1} \), or each \( \text{head}_i \), the context-aware residual attention (CtxResAttn) can be formulated as:

\[ \text{CtxResAttn}(Q, K, V, \text{Prev}, \text{PrevLS}(X_{m-1})) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}} + \text{Prev} + \alpha \text{PrevLS}(X_{m-1}))V, \]  

(6)

where the CtxResAttn is our proposed context-aware residual attention module, \( \text{PrevLS}(X_{m-1}) \) is the correlation attention score of previous one sentence before current input sentence \( X_m \), and \( \alpha \) is an interpolation factor to adjust the weight of historical information. The correlation attention score is calculated from pre-Softmax attention scores \( X_{m-1} \) and \( X_m \) as follows:

\[ \text{PrevLS}(X_{m-1}) = \text{LAN}(s(X_{m-1}), s(X_m)), \]  

(7)

where \( \text{LAN} \) is a linear layer, and \( s(X_m) \) is the pre-Softmax attention score of input \( X_m \). Similar to the ResAttn, the new attention scores are applied on multi-head attention and passed over to the next layer.

![Figure 1: Details of the residual attention and context-aware residual attention.](image)

3.3. Conditional Decoder

The decoder of Transformer contains a lot of linguistic information. In this part, we will describe our proposed conditional decoder, which adds the embedded vector to current input vector through an iterative attention block. As shown in Fig. 2, we recursively get the information of the previous vector, and then incorporate it with the current embedded input vector. Supposing we will look backward \( n \) historic sentences when recognizing the \( k \)-th sentence in the conversation, the label texts \( Y_{m-n} \) and \( Y_{m-n+1} \) will be first processed by the Context-previous attention layer (CtxPrevAttn).

\[ \text{CtxPrevAttn}(Y_{m-n}, Y_{m-n+1}) = \text{Attn}(\text{Embed}(Y_{m-n}), \text{Embed}(Y_{m-n+1}), \text{Embed}(Y_{m-n+1})), \]  

(8)

where the Embed is the word embedding layer and Attn is the dot-product attention mechanism, as described in Eq. 2. Next, the output of the attention layer will be sent to a liner layer to get the position information:

\[ \text{Context}(Y_{m-n}, Y_{m-n+1}) = \text{LAN}(\text{CtxPrevAttn}(Y_{m-n}, Y_{m-n+1}), \text{Embed}(Y_{m-n+1})), \]  

(9)

where LAN is a linear transform layer. Then, we get the output of Context\((Y_{m-n}, Y_{m-n+1}, \text{Embed}(Y_{m-n+2}))\) in turn until the current \( m \)-th sentence. Finally, we send the contextual vector to the decoder of Transformer.
Figure 2: Conditional decoder with \( n \) previous sentences contextual information.

### 4. Experiments

#### 4.1. Dataset

We conduct experiments on three dialogue datasets, including HKUST [24], Switchboard [25], DATATANG-dialog, and a lecture dataset TED-LIUM2. Table 1 summarizes the information of each dataset. The 80-dimensional log-Mel filterbank acoustic features plus 3-dimensional pitch features are used as the input features. For each corpus, the detail configurations of our Transformer model are same as ESPnet Transformer recipes [26] \( (Enc = 12, Dec = 6, \text{dim} = 2048, H = 4, \text{d}_{\text{att}} = 256) \). We use 1996 byte-pair-encoding (BPE) tokens [27] as output units for the English corpora Switchboard and TED-LIUM2, and characters for the Mandarin corpora HKUST and DATATANG-dialog.

<table>
<thead>
<tr>
<th>dataset</th>
<th>lang</th>
<th>hours</th>
<th>test_sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switchboard</td>
<td>en</td>
<td>260</td>
<td>callhome/swbld</td>
</tr>
<tr>
<td>TED-LIUM2</td>
<td>en</td>
<td>213</td>
<td>dev/test</td>
</tr>
<tr>
<td>HKUST</td>
<td>ch</td>
<td>200</td>
<td>train_dev/dev</td>
</tr>
<tr>
<td>DATATANG-dialog</td>
<td>ch</td>
<td>176</td>
<td>train_dev/dev</td>
</tr>
</tbody>
</table>

#### 4.2. Experimental Setup

We train the Transformer and Conformer [17] models with the open-source end-to-end speech processing toolkit ESPnet [26, 28]. We use speed perturbation at ratio 0.9, 1.0, 1.1 for all corpora. We also apply SpecAugment [29] for additional data augmentation. Moreover, the Switchboard dataset is trained with more epochs than the other datasets. The baseline results are trained on independent sentence level, without speaker and context information.

We find that random input training, in other words, in the process of model training, the input sentences are not sent into the model in chronological order, will get better results than the time-ordered input. So instead of training with time-ordered input, we keep additional lists of dialog information during training. We send the features of previous sentences to the encoder to get the attention score. Since there is no need for backpropagation, the demand for computing resources is not significantly increased. For the decoder, we prepare two historical sentences, which means \( n = 2 \), for decoding every sentence.

In order to be consistent with the training input, in the decoding stage, we use the decode results of previous sentences to act as the previous text. For the first sentence in dialag, we just repeat this sentence to replace the position of the previous text. We use two previous sentences information in this paper when training models and the context-aware residual attention has a constant \( \alpha \) value of 0.1.

When decoding, we average the best five models based on the validation loss for recognition on HKUST and DATATANG-dialog, while 10 models for Switchboard and TED-LIUM2. Besides, we use Long Short Term Memory Network language models (LSTM LM) to improve the recognition accuracy. The LM consist of four LSTM layers with 1024 units. The LM of HKUST, TED-LIUM2 and DATATANG-dialog are trained with the transcripts of training sets, and the Switchboard language model is trained with additional Fisher texts. ASR performance is measured by character error rate (CER) or word error rate (WER) depending on the language.

#### 4.3. Results and Analysis

The results are shown in Table 2. For each dataset, the proposed method reduces ASR errors compared with the baseline, and the relative error rate reduction is up to 14%. The DATATANG-dialog and TED-LIUM2 datasets get more error rate reduction. We suppose that those two datasets have greater topic coherence, which enhances the learning ability of the model to the corresponding keywords of specific domains.

#### 4.4. Ablation Study

We also conduct ablation experiments on HKUST and DATATANG-dialog datasets. In Table 3, add\_1\_sen means using the proposed conditional decoder with \( n = 1 \), real means using ResAttn and con means using CtcResAttn. Numbers indicate CERs (%).

##### 4.4.1. Conditional Decoder

From the second row and the third row of Table 3, we can find that the attention decoder input with two previous sentences significantly improved the recognition accuracy. However, the attention decoder with one sentence gets negligible improvement and even the negative effect on the DATATANG-dialog dataset. We come to the conclusion that the longer contextual information, especially in the sentences with both sides of a conversation, can help the decoder to learn more language information.

To verify the hypothesis, we perform speaker-dependent and speaker-independent experiments on dataset DATATANG-dialog. We add the previous sentence information of all speakers in the conversation in the speaker-independent training, and add the previous sentence information of current speaker in the speaker-dependent training. We verify the hypothesis in two different input methods, one is the time-order input of the dialogue, and the other one is the shuffle input. In the time-order training, we sort the input by speaker or simply sort by time to distinguish speaker-dependent or not. In the shuffle training, we keep a list of history context additionally, because we cannot get historical information directly based on the shuffled input. We can see from Table 4 that, the results of speaker-dependent are worse than speaker-independent, both in model trained with time-ordered training dataset and shuffle training dataset. We can draw the conclusion that sentences that span both sides are better than the historical information on either side of the conversation.

##### 4.4.2. Context-aware Residual Attention

The residual attention in encoder does not add any multiplication operations to the computational graph and improves the accuracy of the ASR task. Meanwhile, it achieves competitive results on ASR training with only 90% of the number of epochs of the baseline. As shown in the forth row and the fifth row of Ta-
Table 2: Results of proposed method. Numbers indicate WERs (%) for SWITCHBOARD and TED-LIUM2, and CERs (%) for HKUST and DATATANG(DATATANG-dialog)

<table>
<thead>
<tr>
<th></th>
<th>Switchboard</th>
<th>TED-LIUM2</th>
<th>HKUST</th>
<th>DATATANG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>callhome</td>
<td>swbd</td>
<td>dev</td>
<td>test</td>
</tr>
<tr>
<td>Transformer baseline</td>
<td>17.3</td>
<td>8.5</td>
<td>11.2</td>
<td>9.4</td>
</tr>
<tr>
<td>+Proposed method</td>
<td>16.3</td>
<td>8.3</td>
<td>9.6</td>
<td>8.7</td>
</tr>
<tr>
<td>Conformer baseline</td>
<td>15.6</td>
<td>8.4</td>
<td>10.2</td>
<td>9.0</td>
</tr>
<tr>
<td>+Proposed method</td>
<td>15.1</td>
<td>8.0</td>
<td>9.7</td>
<td>8.7</td>
</tr>
</tbody>
</table>

Table 3: Ablation study on HKUST and DATATANG-dialog.

<table>
<thead>
<tr>
<th></th>
<th>HKUST</th>
<th>DATATANG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train_dev</td>
<td>dev</td>
</tr>
<tr>
<td>Transformer Baseline</td>
<td>24.2</td>
<td>23.6</td>
</tr>
<tr>
<td>+add_1sen</td>
<td>24.0</td>
<td>23.5</td>
</tr>
<tr>
<td>+add_2sen</td>
<td>23.8</td>
<td>23.3</td>
</tr>
<tr>
<td>+add_2sen+real</td>
<td>23.6</td>
<td>23.3</td>
</tr>
<tr>
<td>+add_2sen+con</td>
<td>23.5</td>
<td>22.9</td>
</tr>
</tbody>
</table>

Table 4: Speaker information study on DATATANG-dialog. SI is the speaker-independent training and SD is the speaker-dependent training, indicate CERs (%).

<table>
<thead>
<tr>
<th></th>
<th>time-order</th>
<th>shuffle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train_dev</td>
<td>dev</td>
</tr>
<tr>
<td>baseline</td>
<td>SI</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>25.9</td>
</tr>
<tr>
<td>proposed</td>
<td>SI</td>
<td>23.7</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>24.2</td>
</tr>
</tbody>
</table>

Table 5: CER (%) results of i-vector and our proposed method on DATATANG.

<table>
<thead>
<tr>
<th></th>
<th>train_dev</th>
<th>dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>23.9</td>
<td>25.1</td>
</tr>
<tr>
<td>+proposed method</td>
<td>23.0</td>
<td>23.9</td>
</tr>
<tr>
<td>+i-vector</td>
<td>21.3</td>
<td>21.9</td>
</tr>
<tr>
<td>+i-vector+proposed method</td>
<td>20.6</td>
<td>21.2</td>
</tr>
</tbody>
</table>

Figure 3: Attention scores of 3-th decoder layers, a darker color indicates a higher score for the character.

5. Conclusions

In this paper, we design context-aware residual attention to get the contextual information without extra modules and parameters to the encoder. Moreover, the conditional decoder takes the text information from previous speech and improves the ability of the model to capture long contextual information. The experiments on four datasets demonstrated the effectiveness of our method in enhancing the prediction capacity in dialog ASR tasks. We will improve the decoding speed of our method and context-sensitive decoding strategies in our future work.
6. References


