Relative Positional Encoding for Speech Recognition and Direct Translation

Ngoc-Quan Pham¹ Thanh-Le Ha¹ Tuan-Nam Nguyen¹ Thai-Son Nguyen¹ Elizabeth Salesky²
Sebastian Stueker¹ Jan Niehues³ Alex Waibel¹,4

¹Interactive Systems Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany
²Johns Hopkins University, Baltimore, Maryland
³Department of Data Science and Knowledge Engineering (DKE), Maastricht University
⁴Carnegie Mellon University, Pittsburgh PA, USA

ngoc.pham@kit.edu

Abstract
Transformer models are powerful sequence-to-sequence architectures that are capable of directly mapping speech inputs to transcriptions or translations. However, the mechanism for modeling positions in this model was tailored for text modeling, and thus is less ideal for acoustic inputs. In this work, we adapt the relative position encoding scheme to the Speech Transformer, where the key addition is relative distance between input states in the self-attention network. As a result, the network can better adapt to the variable distributions present in speech data. Our experiments show that our resulting model achieves the best recognition result on the Switchboard benchmark in the non-augmentation condition, and the best published result in the MuST-C speech translation benchmark. We also show that this model is able to better utilize synthetic data than the Transformer, and adapts better to variable sentence segmentation quality for speech translation.

Index Terms: speech recognition, speech translation, transformer, relative position encodings

1. Introduction

It is now evident that neural sequence-to-sequence models [1] are capable of directly transcribing or translate speech in an end-to-end approach. A single neural model which directly maps speech inputs to text outputs advantageously eliminates the individual components in non end-to-end or cascaded approaches, while yielding competitive performance [2, 3]. The hybrid approach for speech recognition and the cascaded approach for speech translation may still give the best accuracy in many conditions, but as neural architectures continue to develop, the gap is closing [4].

The Transformer [5] is a popular architecture choice which has achieved state-of-the-art performance for many sequence learning tasks, particularly machine translation [6]. When applied to speech recognition and direct speech translation, this architecture also stands out as the highest performing option for several datasets [7, 8, 9].

The disadvantage of the Transformer is that, its core function – self-attention – does not have an inherent mechanism to model sequential positions. The original work [5] added position information to the word embeddings via a trigonometric position encoding. Specifically, each element in the sequence is assigned an absolute position with a corresponding encoding (a vector similar to embeddings of the discrete variables, but not updated during training). Recent adaptation to speech recognition [8] showed that the base model, extended in depth, is already sufficient for competitive performance compared to other architecture approaches.

Recently, relative positional encoding has become popularized as a consistent reinforcement for the self-attention. Originally proposed by [12] to replace absolute positions by taking into account the relative positions between the states in self-attention, this method has also been formalized to adapt into language modeling [13], which allows the models to capture very long dependency between paragraphs.

In this work, we bring the advantages of relative position encoding to the Deep Transformer [8] for both speech recognition (ASR) and direct speech translation (ST). The resulting novel model maintains the trigonometric position encodings to better scale with longer speech sequences, and is able to model bidirectional positions as well. On speech recognition, we show that this model consistently improves the Transformer on the standard English Switchboard and Fisher benchmarks (on both 300h and 2000h conditions), and, to the best of our knowledge, is the best published end-to-end model without augmentation on these datasets. More impressively, for speech translation, a single model is able to improve the previous best on the MuST-C benchmark [14] by 7.2 BLEU points. While extending to the IWSLT speech translation task, which is very challenging because it requires of generating audio segmentations, we find that the relative model scales much better with the segmentation quality than the absolute counterpart, and can challenge a very strong cascaded model, which has the advantage of additional model parameters, an intermediate re-segmentation component, and more data.

The project ELITR leading to this publication has received funding from the European Unions Horizon 2020 Research and Innovation Programme under grant agreement No 825460.

1This is the closest speech adaptation that does not change or introduce additional layers (e.g. LSTM [10] or TDNN [11]).
2. Model Description

A speech-to-text model for either automatic speech recognition or direct translation transforms a source speech input with \( N \) frames \( X = x_1, x_2, \ldots, x_N \) into a target text sequence with \( M \) tokens \( Y = y_1, y_2, \ldots, y_M \). The encoder transforms the speech inputs into hidden representations \( h_{IN}^X \). The decoder first generates a language model style hidden representation \( h_0^y \) given the previous inputs, then uses the attention mechanism [15] to generate the relevant context \( c_i \) from the encoder states, which is then combined and generate the output distribution \( o_i \).

\[
\begin{align*}
h_{IN}^X &= \text{ENCODER}(x_1, \ldots, x_N) \quad (1) \\
h_i^y &= \text{DECODER}(y_1, y_2, \ldots, y_{i-1}) \quad (2) \\
c_i &= \text{ATTENTION}(h_i^y, h_1, \ldots, h_{i-1}) \quad (3) \\
o_i &= \text{SOFTMAX}(c_i + h_i^y) \quad (4) \\
y_{i+1} &= \text{sample}(o_i) \quad (5)
\end{align*}
\]

2.1. Transformer

The Transformer [5] uses attention as the main network component to learn encoder and decoder hidden representations. Given three sequences of vectorized states consisting of queries \( Q \in \mathbb{R}^{Q \times D} \), keys \( K \in \mathbb{R}^{K \times D} \), and values \( V \in \mathbb{R}^{V \times D} \), attention computes an energy function \( c_{ij} \) between each query \( Q_i \) and each key \( K_j \). These energy terms are then normalized with a softmax function, and then used to take the weighted average of the values \( V \).

\[
\text{Energy}_{ij} = \text{Energy}(H_i + \tilde{P}_i, H_j + \tilde{P}_j) = H_i W_Q K_j^T + H_i W_Q \tilde{K}^T_j P_j^T + P_i W_Q \tilde{K}^T_j P_j^T + A + B + C + D \quad (8)
\]

Equation 8 gives us an interpretation of the function: in which term A is purely content-based comparison between two hidden states (i.e. speech feature comparison), term D gives a bias between two absolute positions. The other terms represent the specific content and position addressing.

The extension proposed by previously [12] and later [13] changed the terms B, C, D so that only the relative positions are taken into account:

\[
\text{Energy}_{ij} = \text{Energy}(H_i, H_j + \tilde{P}_j) = H_i W_Q K_j^T + H_i W_Q \tilde{K}^T_j P_j^T + A + \tilde{B} + \tilde{C} + \tilde{D} \quad (9)
\]

The new term \( \tilde{B} \) computes the relevance between the input query and the relative distance between \( Q \) and \( K \). Term \( \tilde{C} \) introduces an additional bias \( v \) to the content of the key state \( H_j \), while term \( \tilde{D} \) represents the bias to the global distance. Terms \( \tilde{B} \) and \( \tilde{D} \) also have an additional linear projection \( W_R \) so that the positions and embeddings have different projections.

With this relative position scheme, when the two inputs \( H_i \) and \( H_j \) are shifted (for example, having extra noise or silent in the utterance), the energy function stays the same (for the first layer of the network). Moreover, it can also establish certain quite similar to the encoder counterpart, with the self-attention sub-layer to connect the decoder states, and the feed-forward network. There is an addition encoder-decoder attention layer in between to extract the context vectors from the top encoder states. Furthermore, the Transformer uses residual connections boost information from bottom layers (e.g. the input embeddings) to the top layers. Layer normalization [17] plays a supportive role, keeping the norms of the outputs in check, when used after each residual connection.

2.2. Relative Position Encoding in Transformer

Equation 6 suggests that attention is position-invariant, i.e if the key and value states change their order, the output remains the same. In order to alleviate this problem for this content-based model, positional information within the input sequence is represented in a similar manner with the word embeddings. The positions are treated as discrete variables and then transformed to embeddings either using a look-up table with learnable parameters [18] or with fixed encodings in a trigonometric form:

\[
\begin{align*}
    P_i, 2k & = \sin\left(\frac{i}{10000^{2k/D}}\right) \\
    P_i, 2k + 1 & = \cos\left(\frac{i}{10000^{2k/D}}\right) \quad (7)
\end{align*}
\]

When applied to speech input, this encoding is then added to speech input features [8]. The periodic property of the encodings allow the model to generalize to unseen input length. Following the factorization in [13], we can rewrite the energy function in Equation 6 for self-attention between two encoder hidden states \( H_i \) and \( H_j \) to decompose into 4 different terms:

\[
\text{Energy}_{ij} = \text{Energy}(H_i, H_j, H_j) = H_i W_Q K_j^T + H_i W_Q \tilde{K}^T_j P_j^T + P_i W_Q \tilde{K}^T_j P_j^T + A + B + C + D \quad (8)
\]
inductive bias in the data; for example, the average length of silence or applause, given the global and local bias terms.

2.3. Adaptation to speech inputs

For relative position encodings with speech inputs, should we use learnable embeddings or fixed encodings to represent the distance \( \Delta \)? The latter has the clear advantage that it already has the periodic property, and given that speech input can be as long as thousands of frames, the former approach would require a necessary cut-off \([12]\) to adapt to longer input sequences. These reasons make sinusoidal encodings a logical choice.

Importantly, the relative position scheme above was proposed for autoregressive language models, in which the attention has only one direction. For speech encoders, each state can attend to both left and right directions, thus we propose to use positive distance when the keys are to the left \((j < i)\) and negative distance otherwise. As a result, the encodings for \( P_{\Delta} \) and \( P_{-\Delta} \) will have the same \( \sin \) terms while the \( \cos \) terms will have opposite signs, which gives the model a hint to assign different biases to different directions. Implementation wise, it is able to efficiently compute terms \( B \) and \( D \) with the minimal amount of matrix operations. It is necessary to compute \( 2K-1 \) terms \( H_i W_q W_F P_{\Delta}^T \) with \(-K < k < K\) for each query \( H_i \). (For a sequence with \( K \) states, the distance between one state to another \( k \) is always in that range).\(^3\) This is followed by the shifting trick \([13]\) to achieve the required energy terms.

3. Experiments

3.1. Datasets

**ASR** For ASR tasks, our experiments were conducted on the standard English Switchboard and Fisher data under both benchmark conditions: 300 hours and 2000 hours of training data. Our reported test results are for the Hub5 testset with two subsets Switchboard and CalliHome. Target transcriptions are segmented by byte-pair encoding \([19]\) using 10k merges.

**SLT** We split our SLT task into two different subtasks. Many SLT datasets require an auto-segmentation component to splits the audio into sentence-like segments.\(^4\) For end-to-end models, this step is crucial due to the lack of incremental decoding and higher GPU memory requirements. The recent MuSt-C \([14]\) corpus contains segmentations for both training and test set, requiring no extra segmentation component, and so we use its English-German pair serves as our first experimental benchmark. We further carry out experiments on the IWSLT 2019 evaluation campaign data, a superset of MuST-C, where segmentation is not given; here we can compare the effects of variable-quality segmentation on different end2end models, and also compare models to highly competitive tuned cascades. We use the MuST-C validation data for both tasks.

3.2. Setup

Our baselines for all experiments use the Deep Stochastic Transformer \([8]\). We use the relative encoding scheme above for both encoder and decoder to yield relative Transformers.

For ASR, both our baseline Transformer and relative Transformer have 36 encoder and 12 decoder layers with the model size \( D = 512 \) and the feed-forward networks have the hidden layer size of 2048. Dropout is applied with the same mask across time steps \([20]\) with \( P_{\text{Drop}} = 0.35 \) and also directly at the discrete decoder inputs with \( P_{\text{Drop}} = 0.1 \). All models are trained for at most 120000 steps and the reported model parameters are the average of the 10 checkpoints with lowest perplexities on the cross-validation data.

For SLT, the models and the training process are identical to ASR, with the exception that we use 32 encoder layers.\(^5\) Following the curriculum learning intuition that SLT models benefit from pre-training the speech encoder with ASR \([21]\), we first pre-trained the model for ASR with the parallel English transcripts from MuST-C, and then fine-tune the encoder weights and re-initialize the decoder for SLT. This approach enabled us to consistently train our SLT models without divergence (which may happen when the learning rate is too aggressive or the half-precision GPU mode is used).

For all models, the batch size is set to fit the models to a single GPU\(^6\) and accumulate gradients to update every 12000 target tokens. We used the same learning rate schedule as the Transformer translation model \([5]\) with 4096 warmup steps for the Adam \([22]\) optimizer.

3.3. Speech Recognition Results

<table>
<thead>
<tr>
<th>Models</th>
<th>SWB w/SA</th>
<th>CH w/SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23] BLSTM+LFMMI</td>
<td>9.6</td>
<td>19.3</td>
</tr>
<tr>
<td>[24] Hybrid+LSTM</td>
<td>8.3</td>
<td>17.3</td>
</tr>
<tr>
<td>[25] LAS (LSTM-based)</td>
<td>11.2</td>
<td>7.3</td>
</tr>
<tr>
<td>[26] Shallow Transformer</td>
<td>16.0</td>
<td>11.2</td>
</tr>
<tr>
<td>[26] LSTM-based</td>
<td>11.9</td>
<td>9.9</td>
</tr>
<tr>
<td>3 LSTM-based</td>
<td>12.1</td>
<td>9.5</td>
</tr>
<tr>
<td>+SpecAugment +Stretching</td>
<td>-</td>
<td>8.8</td>
</tr>
</tbody>
</table>

We present ASR results on the Switchboard-300 benchmark in Table 1. It is important to clarify that spectral augmentation (dubbed as SpecAugment) is a recently proposed augmentation method that tremendously improved the regularization ability of seq2seq models for speech recognition \([25]\). In better demonstrate the effect of relative attention, we conduct experiments with and without augmentation.

<table>
<thead>
<tr>
<th>Models</th>
<th>SWB w/SA</th>
<th>CH w/SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23] BLSTM+LFMMI</td>
<td>10.9</td>
<td>14.4</td>
</tr>
<tr>
<td>[24] Hybrid+LSTM</td>
<td>9.4</td>
<td>17.1</td>
</tr>
<tr>
<td>Deep Transformer (Ours)</td>
<td>10.2</td>
<td>9.1</td>
</tr>
<tr>
<td>+SpecPerturb</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Deep Relative Transformer (Ours)</td>
<td>10.2</td>
<td>8.9</td>
</tr>
<tr>
<td>+SpecPerturb</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\[^3\]This is commonly seen in IWSLT evaluation campaigns \([4]\).

\[^4\]The SLT data sequences are longer and thus need more memory

\[^5\]Titan V and Titan RTX with 12 and 24 GB respectively
single GPUs showed similar behavior to ours with SpecAugment. Finally, with additional speed augmentation, relative attention is still additive, with further gains of 0.3 and 0.7 compared to our strong baseline.

Table 2: ASR: Comparison on 2000h SWB+Fisher training set and Hub5’00 test sets. Absolute best is bolded, our best is italicized. WER↓.

<table>
<thead>
<tr>
<th>Models</th>
<th>SWB</th>
<th>CH</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23] Hybrid</td>
<td>8.5</td>
<td>15.3</td>
</tr>
<tr>
<td>[27] Hybrid w/ BiLSTM</td>
<td>7.7</td>
<td>13.9</td>
</tr>
<tr>
<td>[28] Dense TDNN-LSTM</td>
<td>6.1</td>
<td>11.0</td>
</tr>
<tr>
<td>END2END</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[29] CTC</td>
<td>8.8</td>
<td>13.9</td>
</tr>
<tr>
<td>[3] LSTM-based</td>
<td>7.2</td>
<td>13.9</td>
</tr>
<tr>
<td>Deep Transformer (Ours)</td>
<td>6.5</td>
<td>11.9</td>
</tr>
<tr>
<td>Deep Relative Transformer (Ours)</td>
<td>6.2</td>
<td>11.4</td>
</tr>
</tbody>
</table>

The experiments on the larger dataset with 2000h follow the above results for 300h, continuing to show positive effects from that relative position encodings. The error rates on those SWB and CH decrease from 6.5 and 11.9 to 6.2 and 11.4 (Table 2). Our best model is significantly better than previously published CTC [29] and LSTM-based [3] models, and approaches the heavily tuned hybrid system [28] with dense TDNN-LSTM. It is likely possible to reach better error rates, with the help of ensembled models, further data augmentation, and language models. Our experiments here, however, show that the novel relative model is consistently better than the baseline, regardless of the data size and augmentation conditions.

3.4. Speech Translation Results

Our first SLT models were trained only on the MuST-C training data and the results are reported on the COMMON testset\(^7\), using the provided with the segmentation of each utterance which has a corresponding translation. For each utterance, we can directly translate with the end2end model, and the final score can be obtained using standard BLEU scorers such as SacreBLEU [30] because the output and the reference are already sentence-aligned in a standardized way.

Table 3: ST: Translation performance in BLEU↑ on the COMMON testset (no segmentation required).

<table>
<thead>
<tr>
<th>Models</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST: Transformer</td>
<td>18.0</td>
</tr>
<tr>
<td>+SpecAugment</td>
<td>19.3</td>
</tr>
<tr>
<td>+Additional Data [36]</td>
<td>23.0</td>
</tr>
<tr>
<td>Deep Transformer (w/ SpecAugment)</td>
<td>24.2</td>
</tr>
<tr>
<td>+Additional Data</td>
<td>29.4</td>
</tr>
<tr>
<td>Deep Relative Transformer (w/ SpecAugment)</td>
<td>25.2</td>
</tr>
<tr>
<td>+Additional Data</td>
<td>30.6</td>
</tr>
</tbody>
</table>

We experimented with several audio segmentation methods and see that the cascade is less affected by the segmentation quality than the end-to-end models. The results in Table 4 compare two different segmentation methods, LIUM [34] and VAD [35], and four different test sets. The relative Transformer unsurprisingly consistently outperforms the Transformer, regardless of segmentation. Moreover, comparing between the segmenters, the relative model more effectively uses higher segmentation quality, yielding a larger BLEU difference. While the base Transformer only increases up to 0.5 BLEU with better segmentation, this figure becomes up to 2.4 BLEU points for the relative counterpart. In the end, the cascade model still shows that heavily tuned separated components, together with an explicit text segmentation module, is an advantage over end-to-end models, but this gap is closing with more efficient architectures.

4. Conclusion

Speech recognition and translation with end-to-end models have become active research areas. In this work, we adapted the relative position encoding scheme to speech Transformers for these two tasks. We showed that the resulting novel network provides consistent and significant improvement through different tasks and data conditions, given the properties of acoustic modeling. Inevitably, audio segmentation remains a barrier to end-to-end speech translation; we look forward to future neural solutions.

\(^7\)MuST-C is a multilingual dataset and this testset is the commonly shared utterances between the languages.

\(^8\)Available from the evaluation campaign at https://sites.google.com/view/iwslt-evaluation-2019/speech-translation
5. References


