Streaming Automatic Speech Recognition with the Transformer Model

Niko Moritz, Takaaki Hori, Jonathan Le Roux

ICASSP
May, 2020

MITSUBISHI ELECTRIC RESEARCH LABORATORIES (MERL)
Cambridge, Massachusetts, USA
http://www.merl.com
Motivation

• End-to-end automatic speech recognition (ASR) has greatly simplified the pipeline for building and applying ASR systems.

• Offline end-to-end ASR systems have shown to surpass the performance of traditional hybrid DNN-HMM solutions.

• Streaming end-to-end architectures are still lacking behind this success.

• Encoder-decoder based architectures have demonstrated to achieve the best end-to-end ASR results but are difficult to apply in a streaming fashion.

This work

• Our proposed triggered attention (TA) concept is used to overcome these difficulties.

• The TA concept is applied to the transformer architecture, achieving SOTA streaming end-to-end ASR results.
Outline

• Encoder-Decoder Neural Networks
  – Attention
  – Transformer
  – Self-attention
  – Time-Restricted Self-Attention
  – Streaming Encoder-Decoder Attention (prior work)

• Triggered Attention
  – Architecture
  – Frame-Synchronous Decoding Algorithm

• LibriSpeech Results
Attention

Input sequence:

Query vector:

Weight distribution

Embedding/Feature vector:

Output vector:
Encoder-Decoder Attention

Hello  World  <eos>

state

Decoder

Encoder States

Encoder

Acoustic Features

Audio waveform

Feature Extraction
Transformer Architecture

Encoder

Audio Input

Feature Extraction

Positional Encoding

Add & Norm

Feed Forward

Add & Norm

Multi-Head Attention

Self-attention

Output Embeddings

Previous Output

Add & Norm

Encoder-decoder attention

Decoder

Add & Norm

Feed Forward

Add & Norm

Multi-Head Attention

Self-attention

Output

Softmax

Linear

E=12

D=6
Self-Attention

Output sequence:

Input sequence:

current frame
Self-Attention

Input sequence:

- past frames
- current frame
- future frames

Time
Time-Restricted Self-Attention

Output sequence:

Input sequence:

Algorithmic delay: \(\#\text{layers} \cdot \epsilon_{\text{enc}} = 4\) frames

Layer 1

Layer 2

Current frame

Look-ahead \(\epsilon_{\text{enc}} = 2\) frames

Look-ahead \(\epsilon_{\text{enc}} = 2\) frames

\(\epsilon_{\text{enc}} = 2\) frames
Streaming Encoder-Decoder Attention (prior work)

Adaptive Chunking based on Selection Probability

Example:
- Monotonic Chunkwise Attention (MoChA) [1]

Problems:
- Backpropagation with discrete decisions is not possible.
- No frame-synchronous decoding algorithm.
- Detecting word or word-piece positions is a good part of the ASR job that defines insertion and deletion errors.

Triggered Attention (TA) Architecture

Decoding output: \( Y = (y_1, \ldots, y_l) \rightarrow \)

Encoder output: \( X_E = (x_1^E, \ldots, x_N^E) \rightarrow \)

Acoustic features: \( X = (x_1, \ldots, x_T) \rightarrow \)

Language Model

CTC

Triggered Attention Decoder

Joint Decoding

shallow fusion

Transformer model
- \#encoder layers \( E = 12 \)
- \#decoder layers \( D = 6 \)
- attention dimension = 512
- \#attention heads = 8

Frame-Synchronous Decoding

Frame-synchronous CTC prefix beam search [1]:

\[
\ell_1 = (\text{sos}, \text{Hello}, \text{Word}) \quad \ell_7 = (\text{sos}, \text{Hey, World}), \quad \ell_8 = (\text{sos}, \text{Hey, Word}) \quad \ell_3 = (\text{sos}, \text{Hey, Hello})
\]

Set of prefix sequences after pruning:

\[\Omega = \{\ell_1, \ell_2, \ell_3, \ell_7, \ell_8\}\]

\[
\log p(\ell | X_{1:n}^E) = \log p_{\text{prfx}}(\ell | X_{1:n}^E) + \alpha \log p_{\text{LM}}(\ell) + \beta |\ell|
\]

\(\ell\): prefix sequence
\(X_{1:n}^E\): Encoder state sequence for frame \((1, \ldots, n)\)
\(p_{\text{prfx}}\): CTC prefix probability
\(p_{\text{LM}}\): Language model (LM) probability
\(\alpha\): LM weight
\(\beta\): insertion bonus weight
\(|\ell|\): prefix sequence length

Frame-Synchronous Decoding

Frame-synchronous one-pass TA decoding [1]:

$$\log p_{\text{joint}}(\ell|X^{E}_{1:n}) = \lambda \log p_{\text{prfx}}(\ell|X^{E}_{1:n}) + (1 - \lambda) \log p_{\text{ta}}(\ell|X^{E}_{1:n}) + \alpha \log p_{\text{LM}}(\ell) + \beta |\ell|$$

- $p_{\text{ta}}$: Triggered attention probability
- $\lambda$: CTC weight
- $\nu = n' + \varepsilon_{\text{dec}}$
- $n'$: trigger frame
- $\varepsilon_{\text{dec}}$: decoder look-ahead

# LibriSpeech Word Error Rates (WERs) [%]

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Full-sequence CTC-attention decoding [1,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clean</td>
</tr>
<tr>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>Full-sequence</td>
<td>2.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time-restricted encoder</th>
<th>Frame-synchronous CTC prefix beam search</th>
<th>TA: $\varepsilon_{dec} = 18$, delay: $\varepsilon_{dec} \cdot 40 \text{ ms} = 720 \text{ ms}</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_{enc}$ / delay*</td>
<td>Clean</td>
<td>Other</td>
</tr>
<tr>
<td>0 / 30 ms</td>
<td>3.3</td>
<td>3.7</td>
</tr>
<tr>
<td>1 / 510 ms</td>
<td>3.0</td>
<td>3.3</td>
</tr>
<tr>
<td>2 / 990 ms</td>
<td>2.9</td>
<td>3.1</td>
</tr>
<tr>
<td>3 / 1470 ms</td>
<td>2.8</td>
<td>2.9</td>
</tr>
<tr>
<td>Full-sequence</td>
<td>2.5</td>
<td>2.8</td>
</tr>
</tbody>
</table>

* Algorithmic encoder delay: $E \cdot \varepsilon_{enc} \cdot \text{frame-rate} + \text{CNN-delay}$

$E = 12$, frame-rate = 40 ms, CNN-delay = 30 ms


Conclusions

- The triggered attention (TA) concept enables frame-synchronous decoding with an encoder-decoder based model for the first time.
- The TA concept enables joint scoring of an CTC and attention-based decoder model in a streaming fashion.
- The proposed system achieves state-of-the-art results for streaming end-to-end ASR on the LibriSpeech corpus.