Google

Aligner-Encoders:
Self-Attention Transformers can be Self-Transducers

Adam Stooke, Rohit Prabhavalkar, Khe Chai Sim, Pedro Moreno Mengibar

Google Speech

NeurIPS 2024 – Poster 93383

The Problem: Speech-to-Text Alignment in ASR

- Audio input representation typically a much longer sequence than text output.
- Rate of speech can vary widely.

How to bring information from wherever it is in the input sequence to where it belongs in the output sequence?

CTC (2006)

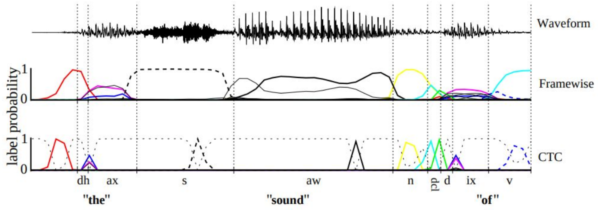
- Dynamic programming to marginalize over all possible alignments in the loss.
- Tokens *in*dependent:
  - “Decode” all embedding frames separately.
  - Post-process out “blanks” & repeats.
- Encoder outputs spiky signals at same timing as inputs.

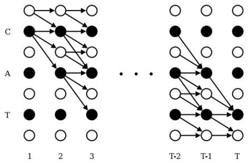
RNN-T (2012)

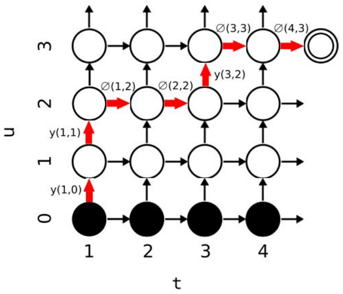
- Dynamic programming to marginalize over all possible alignments in the loss.
- Tokens *inter*dependent:
  - Autoregressive decoding.
  - Decoding lattice tabulates token and timing probabilities.
- Encoder outputs fairly spiky signals at same timing as inputs.

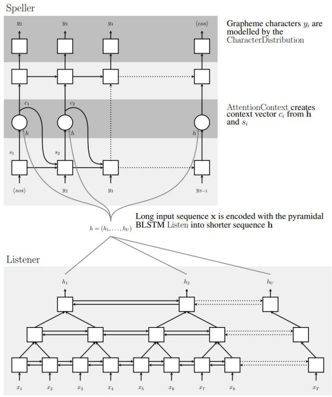
AED / LAS (2015)

- Attention-based decoder accesses entire encoded sequence at every step during autoregressive decoding.
- Encoder outputs (compressed) signals at time corresponding to inputs.



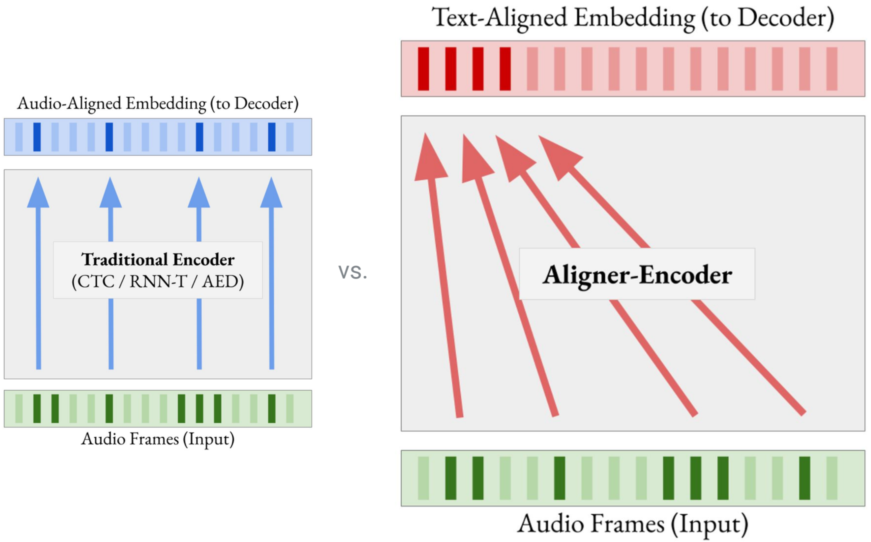






Google

Can the Transformer Encoder do Alignment? — Yes!



Aligner-Encoder — a Simpler ASR Model:

- Simple frame-wise cross-entropy loss of AED.
  (No dynamic programming!)
- Light-weight, text-only recurrence of RNN-T decoder.
  (No cross-attention to all encoder embeddings!)
- Decoding procedure (after encoder computation):
  - Access one embedding frame at a time, in order.
  - Output one token per embedding frame, auto-regressively.
  - Halt at <EOS> prediction. (No “blank” tokens!)
  - Decoding complexity lower than previous models.

Audio: $\mathbf{x} = (x_1, x_2, \ldots, x_T)$
Text: $\mathbf{y} = (y_1, y_2, \ldots, y_U)$

$$\mathbf{h} = f_{\text{enc}}(\mathbf{x})$$
$$g_i = f_{\text{pred}}(g_{i-1}, y_{i-1}), \quad i \leq U$$
$$P(y_i|\mathbf{x}, y_{<i}) = f_{\text{joint}}(h_i, g_i), \quad i \leq U$$

$$\mathcal{L}_{Aligner}(\theta) = -\sum_{i=1}^{U} \log P(y_i|\mathbf{x}, y_{<i}; \theta)$$

LibriSpeech Results – Competitive WER, Faster Inference

WER % (↓)

	DEV	TEST-CLEAN	TEST-OTHER
CTC	2.6	2.8	6.4
RNN-T	2.1	2.1	4.6
AED	2.2	2.4	5.5
ALIGNER	2.2	2.3	5.1

Compute Times (↓)

(MILLISECONDS)	AED	RNN-T	ALIGNER
TRAINING STEP: (ENCODER=560MS)			
DECODER+LOSS	31	290	29
TOTAL	591	850	589
INFERENCE: (ENCODE=32MS; T=300,U=100)			
DECODE STEP	8.5	0.19	0.19
DECODE	850	76	19
TOTAL	832	108	51

- Encoder (all): 17-Layer Conformer (~100M Params).
- Word Error Rate:
  - RNN-T (SOTA) still slightly ahead.
  - Aligner-Encoder is remarkably close→effective ASR.
- Inference Compute Time:
  - **2x** faster than RNN-T.
  - **16x** faster than AED.
  - Auto-regressive computation, but as little as possible.

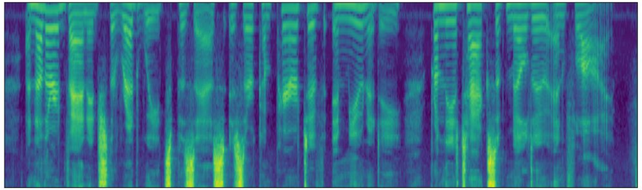
Alignments

Alignment process visible in self-attention weights: “self-transduction”.

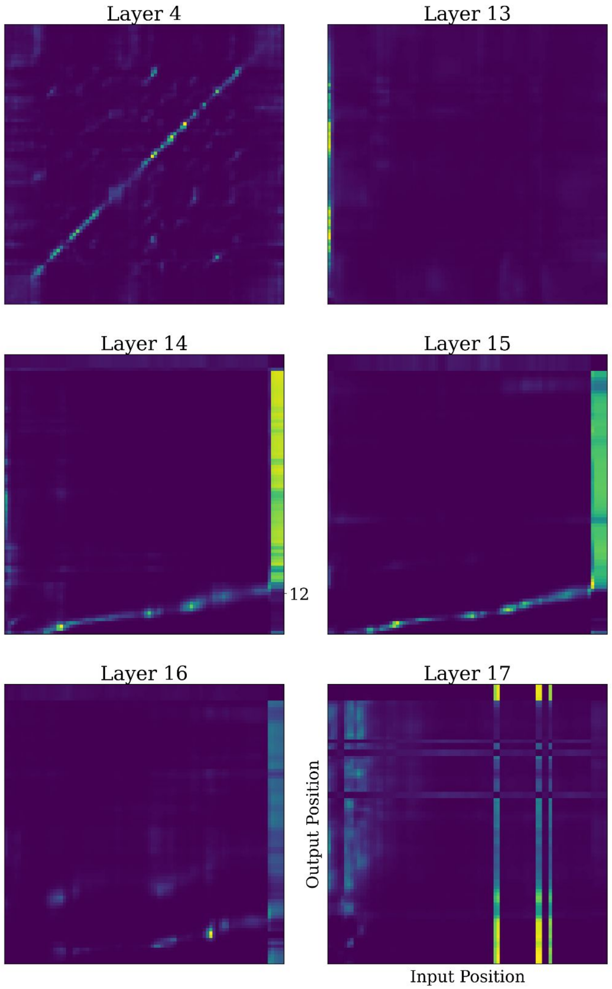
It happens suddenly, in one of the later layers.

Start/end frames possibly used for “bookkeeping”, where usually is silence.

First time alignment has been done fully inside the encoder, before any (auto-regressive) decoding starts.



"illustration italian millet" (12 word-pieces)



Google

Thank you!

More experiments in the paper and at the poster!