Transducer-Based Streaming Deliberation For Cascaded Encoders

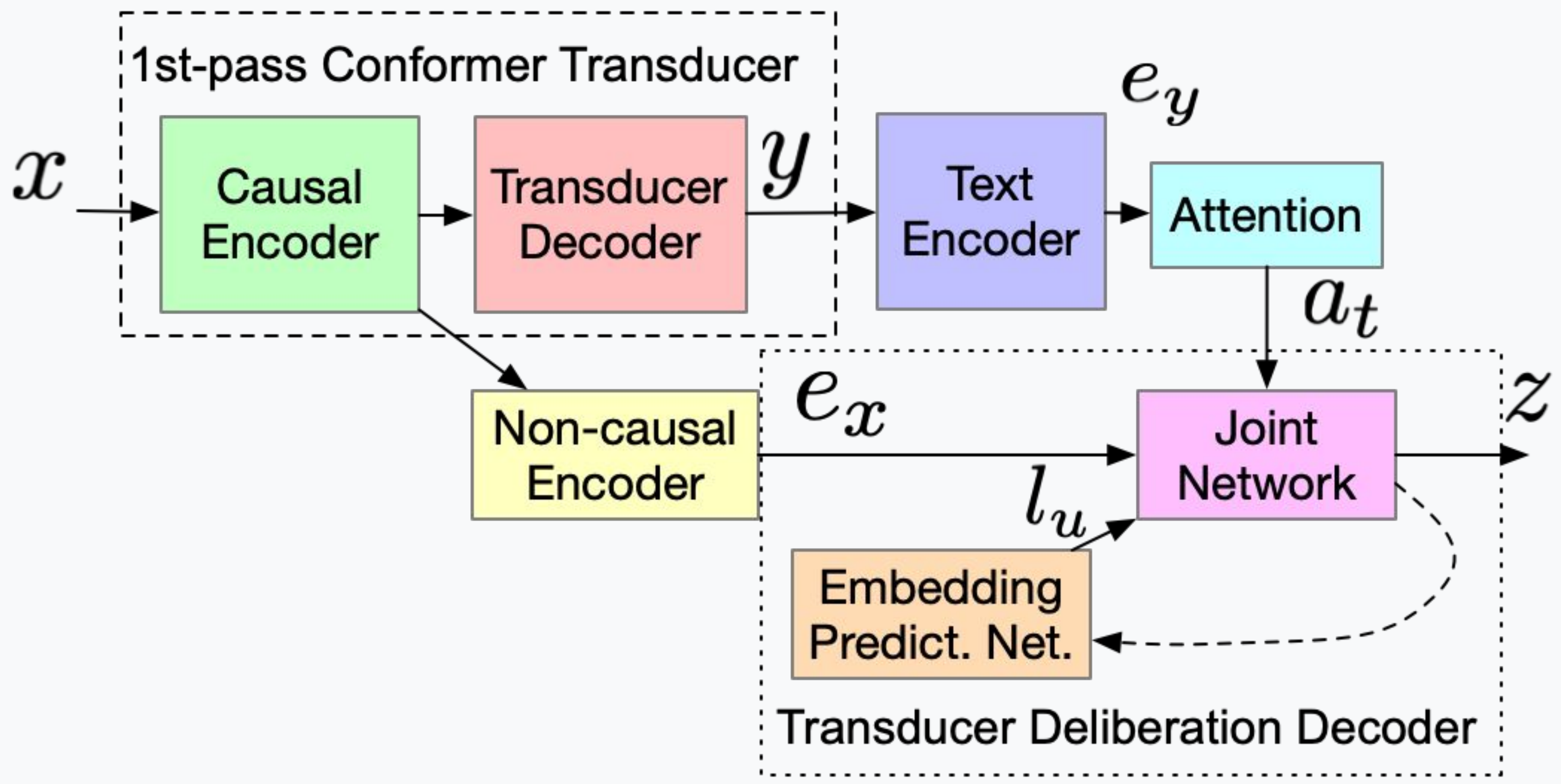
Ke Hu, Tara N. Sainath, Arun Narayanan, Ruoming Pang, Trevor Strohman, Google LLC, USA

ICASSP
Paper ID: 1891

1. Introduction

- Deliberation models improve ASR as 2nd pass rescoring or decoding [1, 2]
  - 1st pass is typically an RNN-T
  - However, 2nd pass deliberation is often based on attention decoders and does not stream
- **Streaming Deliberation**
  - Streaming models are user friendly
  - We use a Transducer Decoder for deliberation
    - 1st-pass hypotheses streamed by a transducer
    - 2nd-pass attends to streamed partial 1st-pass hypotheses, and feed to 2nd-pass joint layer as an additional input
  - The whole model naturally streams
- Novelty
  - Encode first-pass results as a context for 2nd pass decoding
  - Incremental processing instead of requiring full-context for deliberation

2. Model



- **1st Pass** is a conformer transducer
- **2nd Pass**
  - **Non-Causal Encoder**: Conformer layers with right-context for audio
  - **Text Encoder & Attention**
    - Beam search decoding by first-pass conformer transducer
    - Encode output text sequences and compute attention incrementally
  - **Transducer Decoder**: Combine encoder output ($e_x(t)$), prediction network output (l_u), and attention (a_t)

$$c_t = \mathrm{Merge}(a_t, e_x(t)) \qquad h_{t,u} = \tanh(W_{ch}c_t + W_{lh}l_u + b_h)$$

3. Streaming & Latency

- 1st pass naturally streams because it is a transducer
- 2nd pass
  - **Non-Causal Encoder** streams with a latency equal to right-context *R*
  - **Deliberation**: Encode 1st pass hypothesis incrementally and attend to partial sequences

1. Text encoder & Attention

- Use a right-context conformer as text encoder
- For *i*th non-blank token, the right-most frame we need to encode is: $r_i = min(t'_{i+R}, T')$

R: conformer right-context T': maximum time frame of any token
t'_{i+R}: time frame of the non-blank token distance *R* to the right of *i*

2. Attend to partial hypotheses

- At time *t*, only look ahead *A* frames to get a partial sequence:

$$e_y(t) = \{e_{y,k} | \text{ where } r_k < t + A \text{ and } k < L\}$$

A: attention lookahead r_k: right-most frame to encoder *k*th token
L: maximum number of non-blank tokens

3. Transducer decoder naturally streams

Overall latency is from *R* and *A* (choose A=R and parallelize)

4. Experiment Details

- Model Architecture
  - Based on conformer transducer with cascaded encoders [3]
    - 17 causal + 4 non-causal layers (2.88s right context)
  - Deliberation
    - Text encoder: 2-layer 640-D conformer (2.88s right context)
    - Joint layer: Sum encoded audio, prediction network output, and attention
- Inputs and Outputs
  - 32-ms window with 10-ms frame rate
  - Stack previous 3 frames to form 512-D log-Mel features and downsample to 30ms rate
  - Outputs to predict 4,096 lowercase wordpieces
- Datasets
  - Training: ~400k hours from multiple domains
  - Test Sets
    - Short-from (15K utts)
    - TTS utterances: App (16K), Song (15K), Contacts (15K)
      - Contain proper nouns such as app names, song names, and personal contact names

5. Results

Comparison of different E2E models.

Model		WER (%)					
		GVS	CVS	App	Song	Contacts	Avg.
B1	Conf-T	6.8	24.2	17.2	14.8	38.8	20.4
B2	Cascaded encoders	5.4	23.8	10.6	11.5	27.3	15.7
E1	Dual decoder	5.3	23.8	12.8	11.9	27.1	16.1
E2	Cascaded deliberation	**5.4**	**22.2**	**9.7**	**10.3**	**24.7**	**14.5**
	rel. reduct. v.s. B2	*0%*	*-6.7%*	*-8.5%*	*-10.4%*	*-9.5%*	*-7.6%*

Model	Cascaded Encoder	Deliberation
Wins	okay edwards express open	okay adwords express open
	check google accents	check google adsense
Losses	open up the wear os phone	open up the wearos phone
	open up stress meditation	open up stress meditation okay

RWER computed by removing the top English words, and then compute error rates

Model	B2	E2
RWER (%)	8.3	**8.0**

6. Conclusion

WER

- Transducer deliberation improves WER by 3.6% - 10.4% for various long-tail scenarios compared to cascaded encoder [2]

Latency

- The model does not introduce extra latency on top of the cascaded encoder

References

[1] Hu et al., Two-pass deliberation, 2020
[2] Hu et al., Transformer deliberation, 2021
[3] Narayanan et al., Cascaded encoders, 2021