1-2-8-ASR

Samsung Research

Conformer-based On-device Streaming Speech Recognition with KD compression and Two-pass Architecture

Jinhwan Park[1*], Sichen Jin[1*], Junmo Park[1*], Sungsoo Kim[1*], Dhairya Sandhyana[1], Changheon Lee[2], Myoungji Han[2], Jungin Lee[2], Seokyeong Jung[2], Changwoo Han[1], Chanwoo Kim[1]

[1]Samsung Research, South Korea, [2]AI R&D Group, Samsung Electronics, South Korea

* Equally Contributed

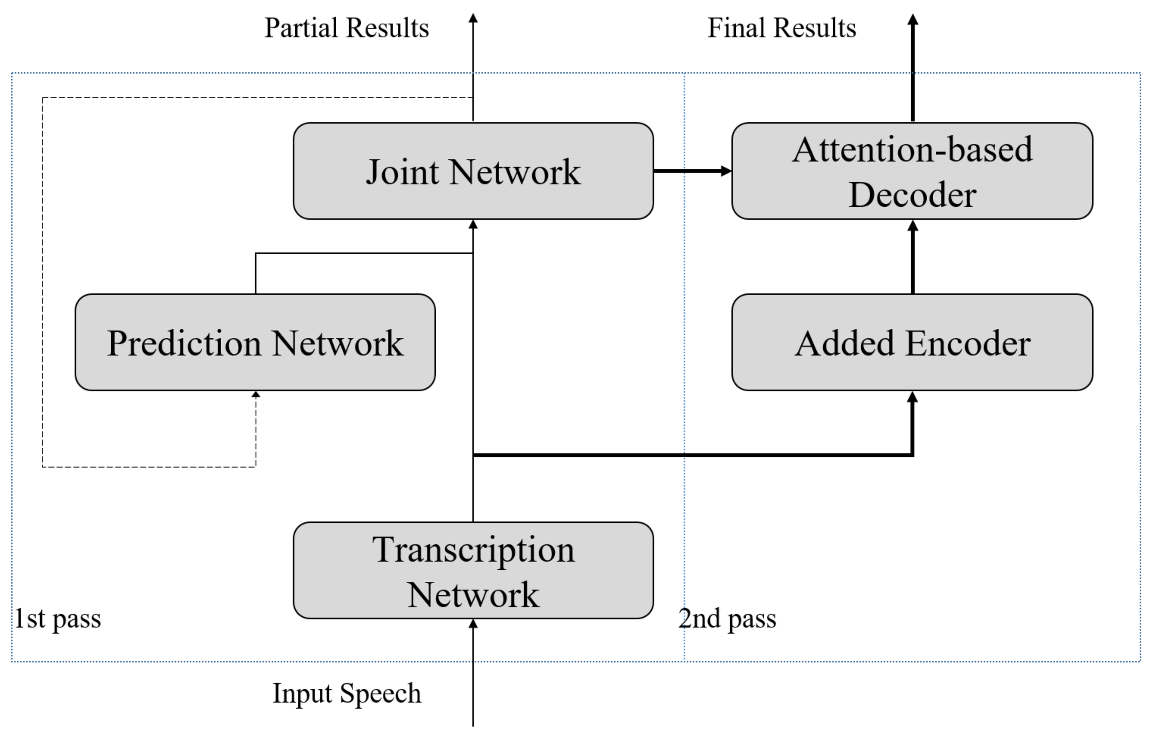
Abstract

- We introduce our two-pass on-device automatic speech recognition (ASR) system which consists of a causal Conformer-transducer as the first pass and a full-context attention model as the second pass.
- The full-context second pass model was compressed by 35% with a 0.02% absolute loss in word error rate (WER) using knowledge distillation (KD).
- Multiple techniques were used for the deployment of the neural ASR model at the inference level.
  - On-device personal adaptation and spell correction are used to solve the mismatch between the training set and the individual test cases.
  - Some hacks are added to the beam search algorithm in order to handle incorrectly segmented speech.
- As a result, the entire system including the two-pass ASR model and a language model (LM) measures **72MB** after 8-bit quantization. It achieves **39% relative WER improvement** compared to our previous work on the test sets collected from the real users.

Model Structure and Training

Two-pass ASR Structure

- Causal Conformer-transducer (Conformer-T) as the first pass
- Full-context attention-based encoder-decoder as the second pass



Knowledge Distillation

- Since the full-context second-pass decoding must be initiated after the termination of the speech, it is essential to reduce the computational cost of the second pass model to achieve low latency.
- The second pass model is compressed with knowledge distillation where a small student model is learned from the big teacher model.

$$L_{dist} = \sum_u P_T(y_u|x, y_{1:u-1}) \log \frac{P_T(y_u|x, y_{1:u-1})}{P_S(y_u|x, y_{1:u-1})}$$

- We follow the 3-stage training with the following objective functions.
  - First-pass model: L_T (first pass)
  - Two-pass teacher model: $L_T + \lambda_1 L_{2nd}$ (first pass and second pass)
  - Two-pass student model with KD: $L_{2nd} + \lambda_2 L_{KD}$ (first pass frozen)

* λ_1 is 1.0 and λ_2 is 0.001 in this work.

Inference Details

Language Model Fusion

- A language model trained from text-only training data predicts the probability of the next token given the previous tokens. We use shallow fusion to borrow the language information from a language model to aid the ASR performance in either general domain or a specific domain.

$$\log P(y_u|x_{1:t}, y_{1:u-1}) = \log P_{\text{ASR}}(y_u|x_{1:t}, y_{1:u-1}) + \sum_i \lambda_i \log P_{\text{LM}_i}(y_u|y_{1:u-1})$$

  - A Transformer-XL based language model is used as a *general domain LM*. The 14MB model is trained with 7GB Korean corpus using the same tokenization as that used for ASR training.
  - For *on-device biasing*, a 6-gram LM is built within the device over the text corpus of the target domains which are synthesized from prepared data-driven templates and the named entities collected from the device.

Spell Correction

- For more strict restrictions over the named entities (NE) produced from the model, we implemented a *spell correction method* which replaces a wrong NE word in the hypothesis with the closest candidate among the collected NE list from the device (i.e. names from contact list, song titles from music playlists)

Algorithm 1 Multi-level spell corrector.

Require: $\mathcal{O}_{asr}$, the hypothesis from transducer
Require: $\mathcal{N}$, the list of named entity candidates
1: $w \leftarrow PatternMatcher(\mathcal{O}_{asr})$
2: **if** w is non-empty **then**
3: **for each** unit $\in$ {word, phoneme, grapheme} **do**
4: **for each** $c \in \mathcal{N}$ **do**
5: **if** $EditDistance_{\text{unit}}(w, c) > T_{\text{unit}}$ **then**
6: **remove** c **from** $\mathcal{N}$
7:
8: **replace** w **with** $e \in \mathcal{N}$ if $\mathcal{N}$ is not empty

Beam Candidate Filtering

- While decoding unexpected silence audios, where an empty hypothesis is expected, we observed that over beam search steps, the score of the beam that contains the empty sentence gets lower and eventually excluded from the beam candidates.
- To solve this problem, we suggest a filtering algorithm during beam search:
  - Ignore all non-blank outputs when $\log P(blank\_token)$ is over -0.05 (95%)
  - Ignore all non-blank outputs whose log-probability is under -4.5 (1%)

Streaming Energy-based Segmentation

- We propose a streaming energy-based segmentation for long speeches. The moving average of spectrogram energy and its highest value are updated for each frame, and the ASR decoder is reset when the ratio of the current value to the highest value is under 0.2.

Experiments

Experimental Setting

- Data
  - Train: 10k hours of transcribed Korean corpus
  - Valid: Random sample from the train set (1h)
  - Test: Usage data (5809 utterances)
- Input feature
  - 80-dim power-mel filterbank
  - window: 25ms, stride: 10ms
  - Stacking 4 frames, skipping every 2 frames
- Vocabulary: 4k wordpieces
- 3-stage scheduled learning rate:
  - Increased linearly
  - Kept constant for about 10k steps
  - Exponentially decayed every step

Model	Conformer-T
first-pass	
Subsampling Conv.	2x(5, 5, 128)
TransNet	Conformer-M [15]
PredNet	2x640 LSTM
JointNet	640 Dense
second-pass	
AddedEncoder	1x1536 LSTM
AddedDecoder	1x1536 LSTM
EncoderAttention	1536 with 4 heads

Evaluation results

(a) Causal Conformer-T (first pass)
(b) Rescoring the candidates of (a) with full-context attention (two-pass)
(c) KD compression on the second pass
- γ_r: Compression rate on recurrent components
- γ_f: Compression rate on other components

(d) Production-ready model fine-tuned with higher-portioned in-domain (command) data
(e) Without Transformer-XL LM shallow fusion

Model	# Params.	WER
(a) Conformer-T	37M	10.19
(b) (a) + rescoring	102M (65M)	9.78
(c) (b) + KD		
(c-1) $\gamma_r = 0.6, \gamma_f = 0.6$	66M (29M)	9.85
(c-2) $\gamma_r = 0.7, \gamma_f = 0.7$	60M (23M)	9.96
(c-3) $\gamma_r = 0.8, \gamma_f = 0.6$	57M (20M)	9.80
(c-4) $\gamma_r = 0.9, \gamma_f = 0.7$	40M (13M)	10.19
(d) (c-3) + fine-tuning	57M (20M)	**5.65**
(e) (d) w/o TFXL LM	57M (20M)	6.54

N-gram Shallow Fusion and Spell Correction (Contact Domain)

Model	Contact	General
Conformer-T + rescoring	11.36	**5.65**
+ *Biasing*	9.89	5.96
+ *Biasing* + *Spell* (syllable)	7.23	5.99
+ *Biasing* + *Spell* (multi)	**4.42**	6.03

Beam Candidate Filtering

Model	FP ratio	WER
Conformer-T + rescoring	1.00	5.65
+ length norm (grid search)	0.50	5.68
+ candidate pruning	**0.16**	**5.65**

Streaming Energy-based Segmentation

Model	Short	Long
Conformer-T + rescoring	**7.87**	64.34
+ Segment (5 secs)	9.31	24.38
+ Segment (10 secs)	8.03	26.70
+ Segment (energy)	8.04	**13.76**

Real-time Factor (Galaxy Note 10, single core CPU)

Model	xRT
RNN-T + rescoring	0.375
Conformer-T + rescoring	**0.143**

Conclusion

- We constructed an on-device streaming speech recognition solution based on two-pass architecture.
- We applied multiple techniques in both training and inference stage including KD compression, shallow fusion, spell correction, beam candidate filtering and streaming segmentation.
- The proposed ASR system achieved 5.6% WER on the Korean test set with 0.14 in xRT and 72MB in model size.