Supplementary material for "A Purely End-to-end System for Multi-speaker Speech Recognition"

Hiroshi Seki[1,2]*, Takaaki Hori[1], Shinji Watanabe[3], Jonathan Le Roux[1], John R. Hershey[1]
[1]Mitsubishi Electric Research Laboratories (MERL)
[2]Toyohashi University of Technology
[3]Johns Hopkins University

1 Architecture of the encoder-decoder network

In this section, we describe the details of the baseline encoder-decoder network which is further extended for permutation-free training. The encoder network consists of a VGG network and bi-directional long short-term memory (BLSTM) layers. The VGG network has the following 6-layer CNN architecture at the bottom of the encoder network:

Convolution (# in = 3, # out = 64, filter = 3×3)
Convolution (# in = 64, # out = 64, filter = 3×3)
MaxPooling (patch = 2×2, stride = 2×2)
Convolution (# in = 64, # out = 128, filter = 3×3)
Convolution (# in=128, # out=128, filter=3×3)
MaxPooling (patch = 2×2, stride = 2×2)

The first 3 channels are static, delta, and delta delta features. Multiple BLSTM layers with projection layer $\mathrm{Lin}(\cdot)$ are stacked after the VGG network. We defined one BLSTM layer as the concatenation of a forward LSTM $\overrightarrow{\mathrm{LSTM}}(\cdot)$ and a backward LSTM $\overleftarrow{\mathrm{LSTM}}(\cdot)$:

$$\overrightarrow{H} = \overrightarrow{\mathrm{LSTM}}(\cdot) \tag{29}$$
$$\overleftarrow{H} = \overleftarrow{\mathrm{LSTM}}(\cdot) \tag{30}$$
$$H = [\mathrm{Lin}(\overrightarrow{H}); \mathrm{Lin}(\overleftarrow{H})], \tag{31}$$

When the VGG network and the multiple BLSTM layers are represented as $\mathrm{VGG}(\cdot)$ and $\mathrm{BLSTM}(\cdot)$, the encoder network in Eq. (2) maps the input feature vector O to internal representation H as follows:

$$H = \mathrm{Encoder}(O) = \mathrm{BLSTM}(\mathrm{VGG}(O)) \tag{32}$$

The decoder network sequentially generates the n-th label y_n by taking the context vector c_n and the label history $y_{1:n-1}$:

$$y_n \sim \mathrm{Decoder}(c_n, y_{n-1}). \tag{33}$$

The context vector is calculated in an location based attention mechanism (Chorowski et al., 2015) which weights and sums the C-dimensional sequence of representation $H = (h_l \in \mathbb{R}^C | l = 1, \ldots, L)$ with attention weight $a_{n,l}$:

$$c_n = \mathrm{Attention}(a_{n-1}, e_n, H), \tag{34}$$
$$\triangleq \sum_{l=1}^{L} a_{n,l} h_l. \tag{35}$$

The location based attention mechanism defines the weights $a_{n,l}$ as follows:

$$a_{n,l} = \frac{\exp(\alpha k_{n,l})}{\sum_{l=1}^{L} \exp(\alpha k_{n,l})}, \tag{36}$$
$$k_{n,l} = w^{\mathrm{T}} \tanh(V^{\mathrm{E}} e_{n-1} + V^{\mathrm{H}} h_l + V^{\mathrm{F}} f_{n,l} + b), \tag{37}$$
$$f_n = F * a_{n-1}, \tag{38}$$

where w, V^E, V^H, V^F, b, F are tunable parameters, α is a constant value called inverse temperature, and $*$ is the convolution operation. We used 10 convolution filters of width 200, and set α to 2. The introduction of f_n makes the attention mechanism take into account the previous alignment information. The hidden state e is updated recursively by an updating LSTM function:

$$e_n = \mathrm{Update}(e_{n-1}, c_{n-1}, y_{n-1}), \tag{39}$$
$$\triangleq \mathrm{LSTM}(\mathrm{Lin}(e_{n-1}) + \mathrm{Lin}(c_{n-1}) + \mathrm{Emb}(y_{n-1})), \tag{40}$$

where $\mathrm{Emb}(\cdot)$ is an embedding function.

*This work was done while H. Seki, Ph.D. candidate at Toyohashi University of Technology, Japan, was an intern at MERL.

Table 1: Examples of recognition results. Errors are emphasized as capital letter. "_" is a space character, and a special token "*" is inserted to pad deletion errors.

(1) Model w/ permutation-free training (CER of HYP1: 12.8%, HYP2: 0.9%)
HYP1: t h e _ s h u t t l e _ * * * I S _ I N _ t h e _ f i r s t _ t H E _ l i f E _ o * f _ s i n c e _ t h e _ n i n e t e e n _ e i g h t y _ s i x _ c h a l l e n g e r _ e x p l o s i o n
REF1: t h e _ s h u t t l e _ W O U L D _ B E _ t h e _ f i r s t _ t * O _ l i f T _ o F f _ s i n c e _ t h e _ n i n e t e e n _ e i g h t y _ s i x _ c h a l l e n g e r _ e x p l o s i o n
HYP2: t h e _ e x p a n d e d _ r e c a l l _ w a s _ d i s c l o s e d _ a t _ a _ m e e t i n g _ w i t h _ n . _ r . _ c . _ o f f i c i a l s _ a t _ a n _ a g e n c y _ o f f i c e _ o u t s i d e _ c h i c a g o
REF2: t h e _ e x p a n d e d _ r e c a l l _ w a s _ d i s c l o s e d _ a t _ a _ m e e t i n g _ w i t h _ n . _ r . _ c . _ o f f i c i a l s _ a t _ a n _ a g e n c y _ o f f i c e _ o u t s i d e _ c h i c a g o
(2) Model w/ permutation-free training (CER of HYP1: 91.7%, HYP2: 38.9%)
HYP1: I T _ W A S _ L a s t _ r * A I S e * D _ * I N _ J U N E _ N I N E t E e N _ e * I G h T Y _ f I V e _ T O _ * T H I R T Y
REF1: * * * * * * * * a s t * r O N O M e R S _ S A Y _ T H A T _ * * * * t H e * _ e A R T h ' S _ f A T e _ I S _ S E A L E D
HYP2: * * * * a N D _ * * s t * r O N G e R S _ S A Y _ T H A T _ * * * * t H e * _ e * A R t H _ f A T e _ I S _ t o _ f o r t y _ f i v e _ d o l l a r s _ f r o m _ t h i r t y _ f i v e _ d o l l a r s
REF2: I T _ W a * S _ L A s t _ r A I S e * D _ * I N _ J U N E _ N I N E t E e N _ e I G H t Y _ f I V e * * * _ t o _ f o r t y _ f i v e _ d o l l a r s _ f r o m _ t h i r t y _ f i v e _ d o l l a r s

Algorithm 1 Generation of multi speaker speech dataset

$n_{\text{reuse}} \Leftarrow$ maximum number of times same utterance can be used.
$U \Leftarrow$ utterance set of the corpora.
$C_k \Leftarrow n_{\text{reuse}}$ for each utterance $U_k \in U$
for $U_k \in U$ **do**
  $P(U_k) = C_k \,/\, \sum_l C_l$
end for
for U_i in U **do**
  Sample utterance U_j from $P(U)$ while ensuring speakers of U_i and U_j are different.
  Mix utterances U_i and U_j
  if $C_j > 0$ **then**
    $C_j = C_j - 1$
    for $U_k \Leftarrow U$ **do**
      $P(U_k) = C_k \,/\, \sum_l C_l$
    end for
  end if
end for

2 Generation of mixed speech

Each utterance of the corpus is mixed with a randomly selected utterance with the probability, $P(U_k)$, that moderates over-selection of specific utterances. $P(U_k)$ is calculated in the first for-loop as a uniform probability. All utterances are used as one side of the mixture, and another side is sampled from the distribution $P(U_k)$ in the second for-loop. The selected pairs of utterances are mixed at various signal-to-noise ratios (SNR) between 0 dB and 5 dB. We randomized the starting point of the overlap by padding the shorter utterance with silence whose duration is sampled from the uniform distribution within the length difference between the two utterances. Therefore, the duration of the mixed utterance is equal to that of the longer utterance among the unmixed speech. After the generation of the mixed speech, the count of selected utterances C_j is decremented to prevent of over-selection. All counts C are set to n_{reuse}, and we used $n_{\text{reuse}} = 3$.

3 Examples of recognition results and error analysis

Table 1 shows examples of recognition result. The first example (1) is one which accounts for a large portion of the evaluation set. The SNR of the HYP1 is -1.55 db and that of HYP2 is 1.55 dB. The network generates multiple hypotheses with a few substitution and deletion errors, but without any overlapped and swapped words. The second example (2) is one which leads to performance reduction. We can see that the network makes errors when there is a large difference in length between the two sequences. The word "thirty" of HYP2 is injected in HYP1, and there are deletion errors in

HYP2. We added a negative KL divergence loss to ease such kind of errors. However, there is further room to reduce error by making unshared modules more cooperative.

References

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 577–585.