Attention-based End-to-end Speech Recognition on Voice Search

Changhao Shan[1,2], Junbo Zhang[2], Yujun Wang[2], Lei Xie[1]

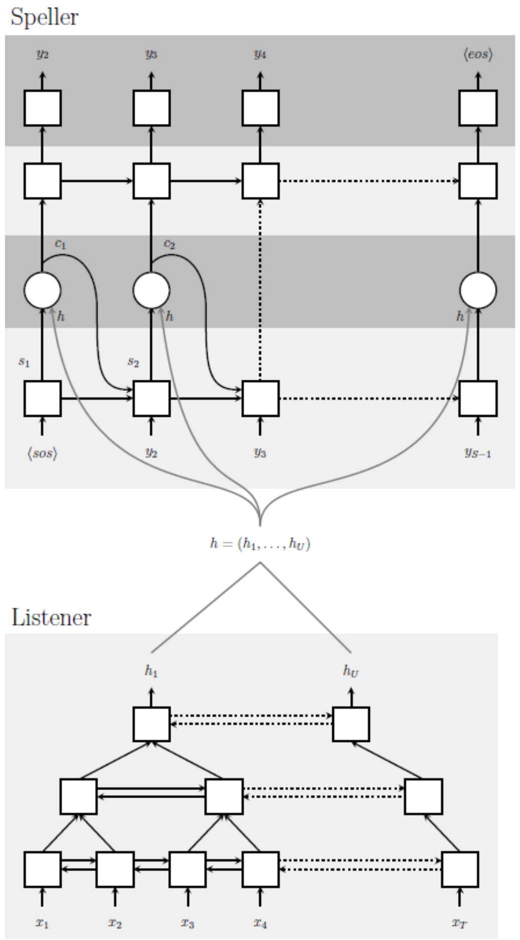
1. Shaanxi Provincial Key Lab of Speech and Image Information Processing, Northwestern Polytechnical University, Xi'an
2. Xiaomi Corporation, Beijing

Outline

- A brief review of LAS
- Train a LAS for Mandarin
  - Embedding
  - Frame skipping
- Attention mechanism
  - Content-based vs. Location-based
  - Attention smoothing
- Decoding
  - Softmax with temperature
  - Language model integration
- Experiment
  - Dataset
  - LAS setup
  - Result
- Conclusion

Listen, Attend and Spell (LAS)

- Note that the "LAS" here is refer to both the works [1] and [2] (they are almost same)
- Listen: or Encoder, extracts higher-level features
  - The Encoder does not require to be pyramidal
- Attend: weights the outputs of the Encoder
- Spell: or Decoder, generates a prediction of characters

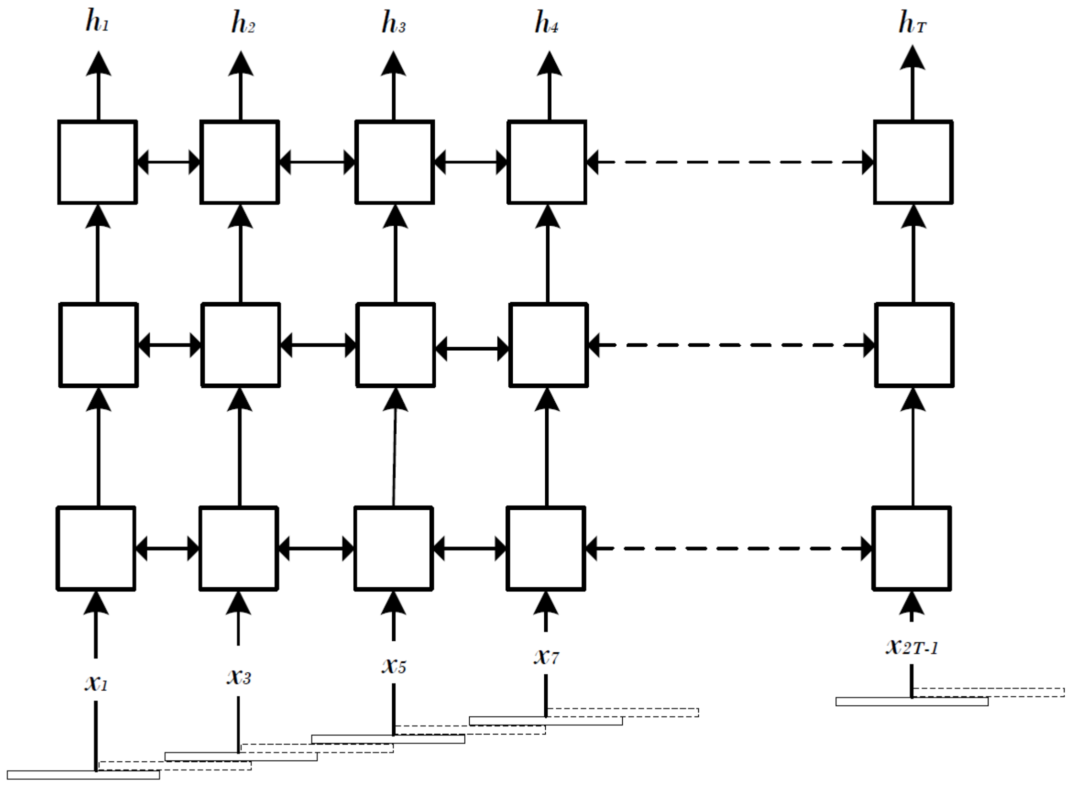


[1] Chan, W., Jaitly, N., Le, Q., & Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, ICASSP 2016.
[2] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. End-to-End Attention-based Large Vocabulary Speech Recognition. ICASSP 2016

The Listen module

- The encoder is normally implemented as a bidirectional recurrent network
- Zero-pad the input feature into a fixed-length
- The Listen module maps input feature $\mathbf{x}$ (padded) into a fixed-length feature representation $\mathbf{h}$

$$\mathbf{h} = EncoderRNN(\mathbf{x})$$



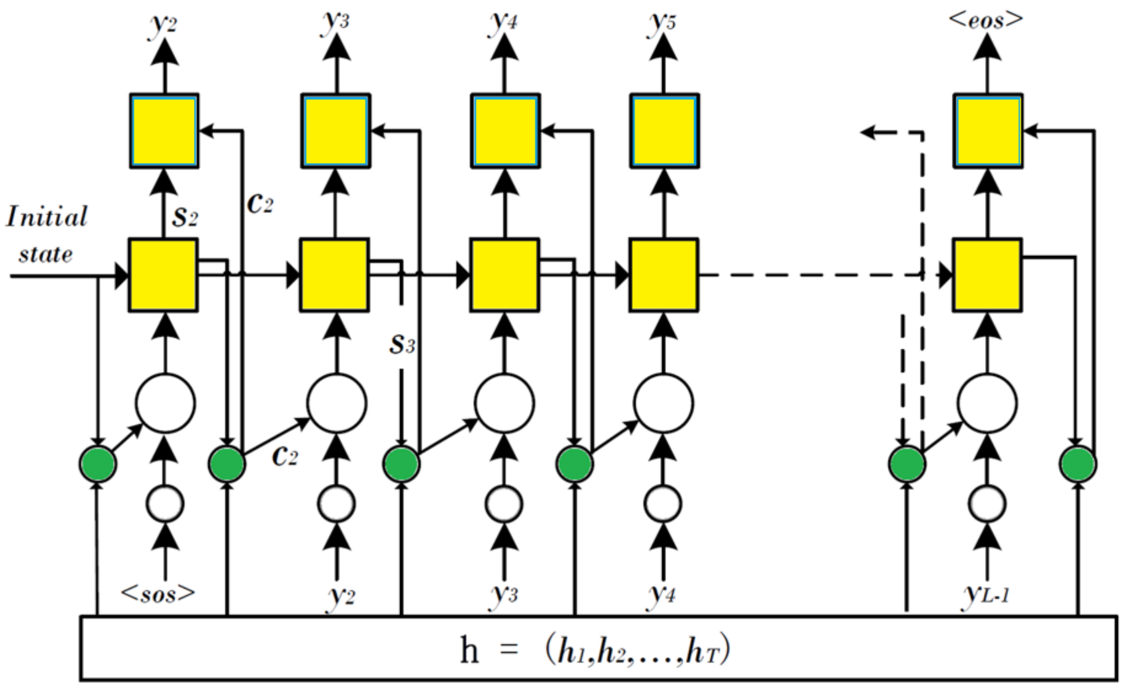
The Attend and Spell module

- The Attend module weights the encoded features $\mathbf{h}$, resulting in a context vector $\mathbf{c}$
- The Spell module (or Decoder) takes the attention context vector $\mathbf{c}$ and the previous prediction to generate a prediction of the next output

$$c_i = AttendtionContext(s_i, \mathbf{h})$$

$$s_i = DecodeRNN([y_{i-1}, c_{i-1}], s_{i-1})$$

$$P(y_i|\mathbf{x}, y_{i-1}) = CharacterDistribution(c_i, s_i)$$



Challenge of training LAS on Mandarin

- The attention model is difficult to converge on Mandarin [1]
  - Many thousands of characters
  - Chinese characters give limited information on the sounds of the spoken language
    - In some work, phonetic representation (pinyin) was introduced to help training

- We aim to train a Mandarin LAS using Chinese characters directly (without pinyin's help)

[1] Chan, W., & Lane, I. On online attention-based speech recognition and joint Mandarin character-pinyin training. Interspeech 2016

Train Mandarin LAS

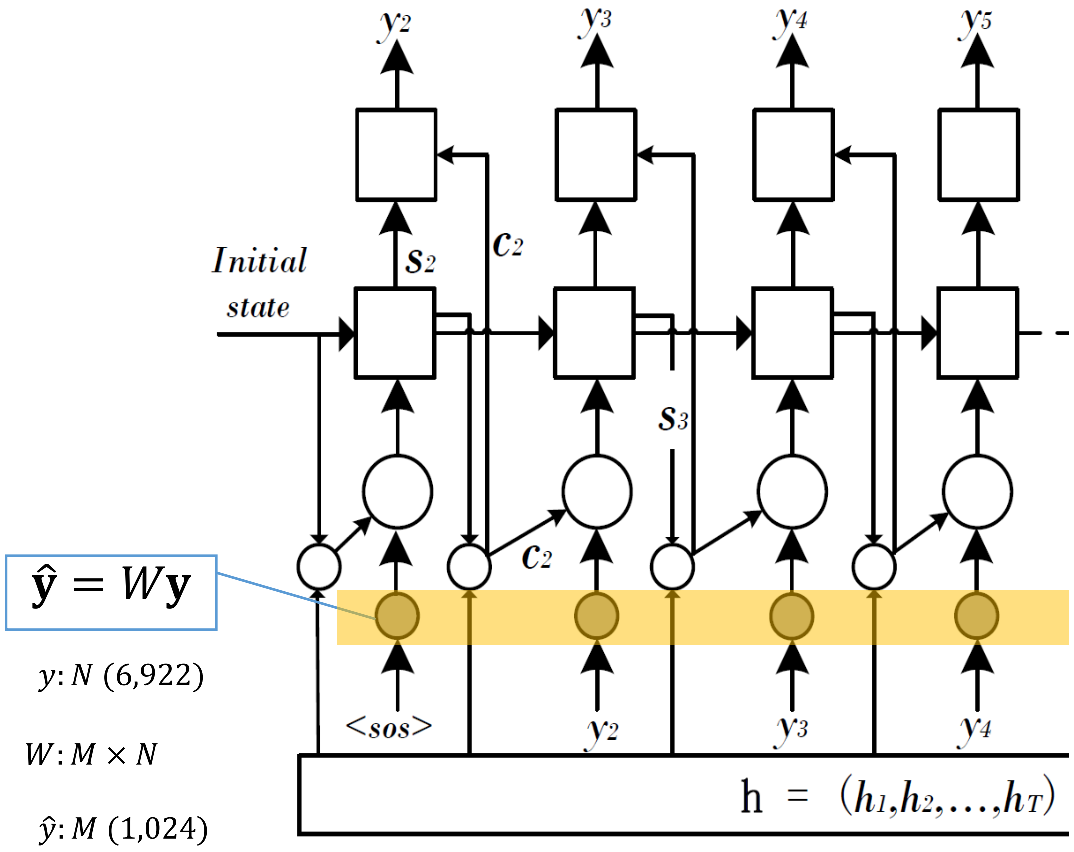
- Tricks we tried but still not converge (CER > 95%)
  - Adjust learning rate
  - Adjust batch size
  - L2 regularization
  - Dropout
  - Adam optimizer
  - Pyramidal encoder [1]

- Worked tricks for converging
  - Character embedding
  - Frame skipping

[1] Chan, W., Jaitly, N., Le, Q., & Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, ICASSP 2016.

Character embedding

- The character embedding layer maps one-hot vectors into embedded vectors
  - dim of the one-hot vector (N): 6,922
  - dim of the embedded vector (M): 1,024
- It is updated in the whole LAS model training procedure
- The i-th row of W is the embedded vector of the character with index i
  - acts as a lookup-table



Frame skipping

- Utterance length of our dataset
  - 900 frames after padding
- Inspired by low frame rate [1]
  - There is no need to assume signal stationarity for RNN
  - Studies shows that low-frame rate not only makes decoding faster, but also improves the accuracy
- Borrowing the low frame rate idea, we do frame skipping in the training of LAS encoder

[1] Pundak, G., & Sainath, T. Lower frame rate neural network acoustic models. INTERSPEECH 2016

Works on attention

- Compared two attention methods
  - Content-based attention
  - Location-based attention

- Attention smoothing

Attention mechanism

- For the i-th step to generates an output y_i :
  - The attention mechanism weights the feature representation $\mathbf{h}$ by the weights α_i to generate a context feature
  - α_i is learned from $\mathbf{h}$ and the Decoder LSTM hidden state s_{i-1}
  - $e_{i,j}$: how well the inputs around position j and the output at position i match

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^{T} \exp(e_{i,j})}$$

$$e_{i,j} = Score(s_{i-1}, h_j)$$

[1] Dzmitry Bahdana, Bahdanau, D., Cho, K., & Bengio, Y. Neural Machine Translation By Jointly Learning To Align and Translate. ICLR 2015

Content-based vs. Location-based

- Content-based attention [1]

$$e_{i,j} = \boldsymbol{\omega}^T \tanh(\mathbf{W}s_{i-1} + \mathbf{V}h_j + \mathbf{b})$$

- Location-based attention [2]

$$e_{i,j} = \boldsymbol{\omega}^T \tanh(\mathbf{W}s_{i-1} + \mathbf{V}h_j + \mathbf{U}\mathbf{f}_i + \mathbf{b})$$

$$\mathbf{f}_i = \mathbf{F} * \alpha_{i-1}$$

$$\alpha_{i,j} = Softmax(e_{i,j})$$

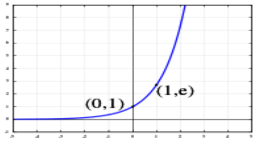
Attention	CER / %	SER / %
Content-based	4.05	9.10
Location-based	3.82	8.17

[1] Dzmitry Bahdana, Bahdanau, D., Cho, K., & Bengio, Y. Neural Machine Translation By Jointly Learning To Align and Translate. ICLR 2015
[2] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. Attention-Based Models for Speech Recognition. NIPS 2015

Inspire of attention smoothing

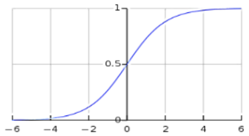
- The α_i distribution is typically very sharp, and thus it focuses on only a few frames of $\mathbf{h}$
- Long context information may be useful for the voice search task
- In the Softmax, the exponential function (unbounded) could be replaced by logistic sigmoid (bounded) [1]

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^{T} \exp(e_{i,j})}$$



exponential

$$\alpha_{i,j} = \frac{\sigma(e_{i,j})}{\sum_{j=1}^{T} \sigma(e_{i,j})} \quad , where \quad \sigma(e_{i,j}) = \frac{1}{1 + \exp(-e_{i,j})}$$



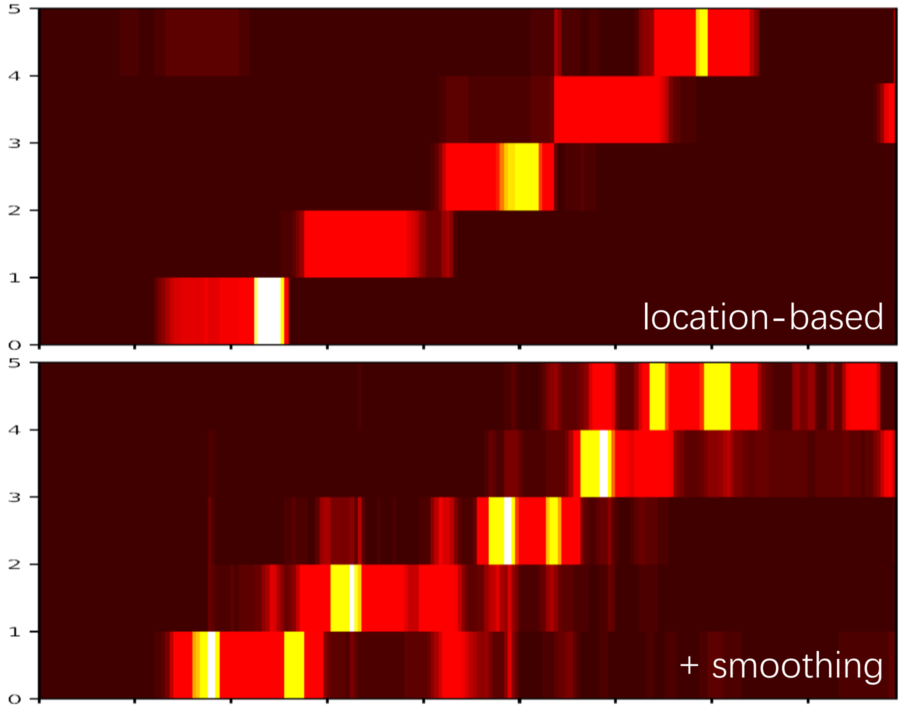
logistic sigmoid

[1] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. Attention-Based Models for Speech Recognition. NIPS 2015

Attention smoothing

- We simply replace Softmax by logistic sigmoid
- Although $\alpha_{i,j}$ is no longer required to sum to 1.0, $\alpha_{i,j}$ do not depends on all $e_{i,j}$ though T
- It makes the attention computing faster

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^{T} \exp(e_{i,j})} \Rightarrow \alpha_{i,j} = \frac{1}{1 + \exp(-e_{i,j})}$$



Attention	CER / %	SER / %
Location-based	3.82	8.17
+ Attention smoothing	**3.58**	**7.43**

Decoding

- We used a simple left-to-right beam search algorithm during decoding [1]
- Temperature [2]
  - hyper-parameter in Softmax to smooth the distribution of characters
  - increasing temperature make the distribution over characters more uniform

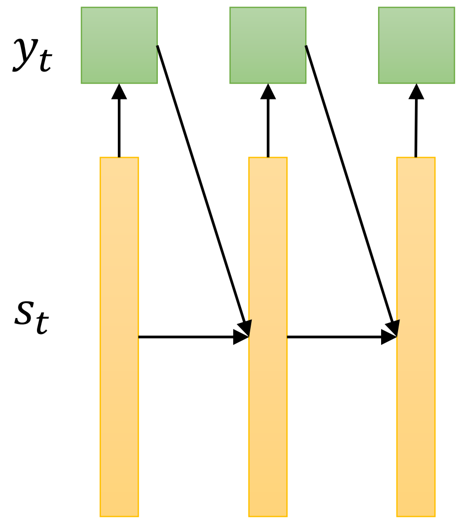
$$p(y_i|\mathbf{x}, y_{i-1}) = \frac{\exp(\frac{o_t}{\tau})}{\sum_j \exp(\frac{o_j}{\tau})}$$

- The temperature might be as another attention smoothing way [3], but we have not yet get a good result with it

[1] Sutskever, I., Vinyals, O., & Le, Q. V. Sequence to Sequence Learning with Neural Networks. Nips 2014
[2] Chorowski, J., & Jaitly, N. Towards better decoding and language model integration in sequence to sequence models. Interspeech 2017
[3] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. Attention-Based Models for Speech Recognition. NIPS 2015

Language model

- The Spell module of the model is an implicit character-level language model
  - predict the next character according the history
- The model itself is insufficient to learn a complex language mode [1]
- The transcripts of the acoustic training data are limit
  - we have huge text data without audio



$s_i \leftarrow (y_{i-1}, c_{i-1}, s_{i-1})$

$y_i \leftarrow s_i$

[1] Chorowski, J., & Jaitly, N. Towards better decoding and language model integration in sequence to sequence models. Interspeech 2017

External language model

- Build a character-level external language model [1]
  - Rewrite an existed word-level n-gram LM as WFST (G)
    - input/output label: word (a word consist of several Chinese characters)
  - Use a WFST (L) to transduce character sequence to word
    - input label: character
    - output label: word
  - Compose L and G to get external character-level LM

$$T = \min(\det(L^\circ G))$$

- Combines the internal and the external language model

$$C = -\sum_i [\log p(y_i|\mathbf{x}, y_{i-1}, \cdots, y_1) + \gamma T]$$

[1] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. . End-to-End Attention-based Large Vocabulary Speech Recognition. ICASSP 2016

Dataset

- 3,000 hours MiTV dataset
- The longest utterance is about 10 secs
- Collected from the microphone on the MiTV remote controller
- Includes 6,922 Chinese characters
- The test set includes 3000 utterances

Experiment setup

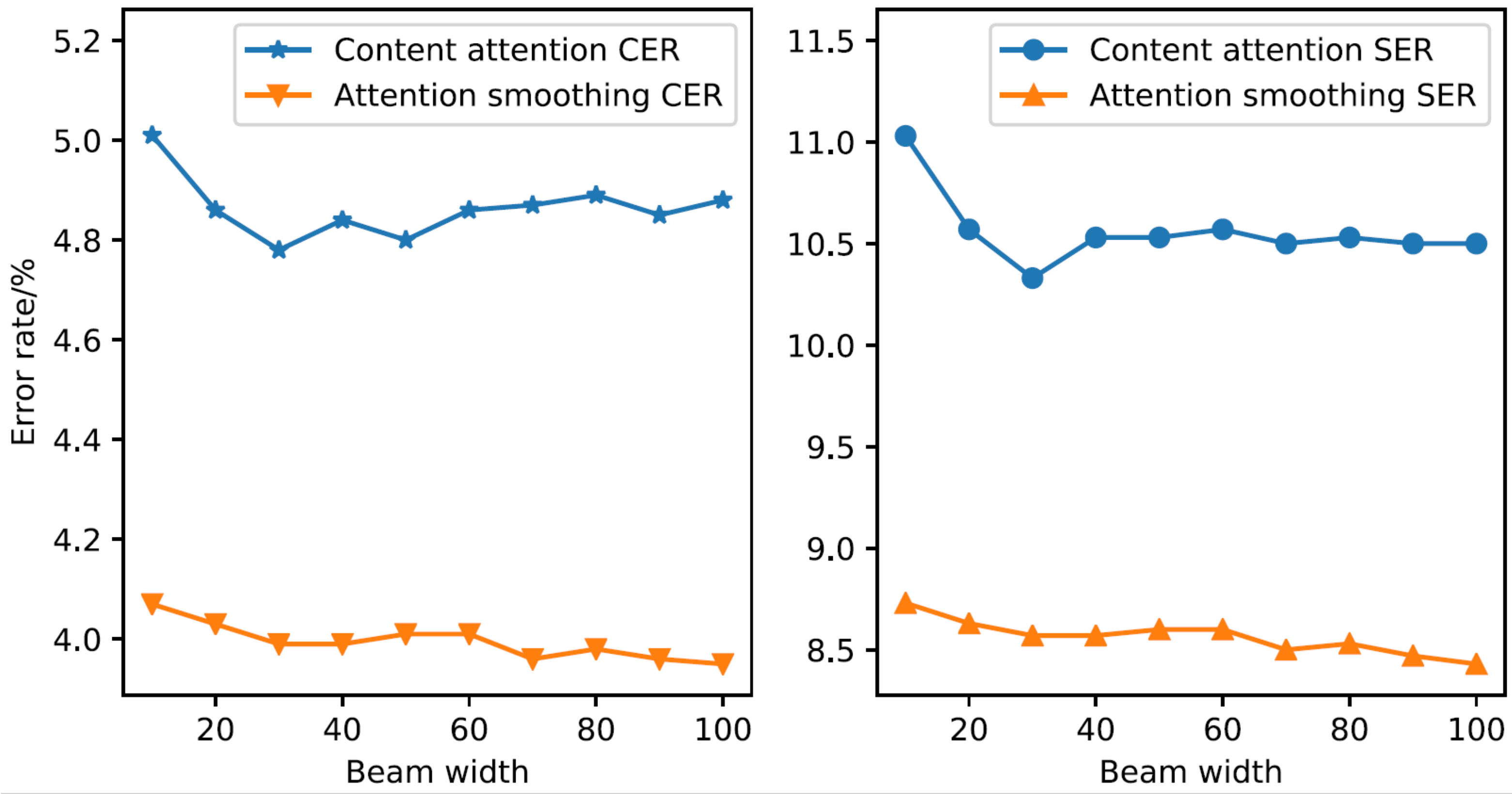
- Acoustic feature
  - 80 Mel-scale filter-bank coefficients
  - delta and delta-delta acceleration
  - mean and variance normalization for each speaker
- Encoder
  - 3-layer BLSTM
  - 512 LSTM units per layer
- Decoder
  - 1-layer LSTM
  - 256 LSTM units
  - 6,925 output labels

Experiment setup

- Hyper-parameters
  - Initialized with the normalized initialization
  - Gradient norm clipping to 1
  - Gaussian weight noise
  - L2 weight decay 1e-5
  - ADAM as the optimization method
  - Learning rate 1e-3 (1e-4 after it converged)
  - Cross-entropy as the cost

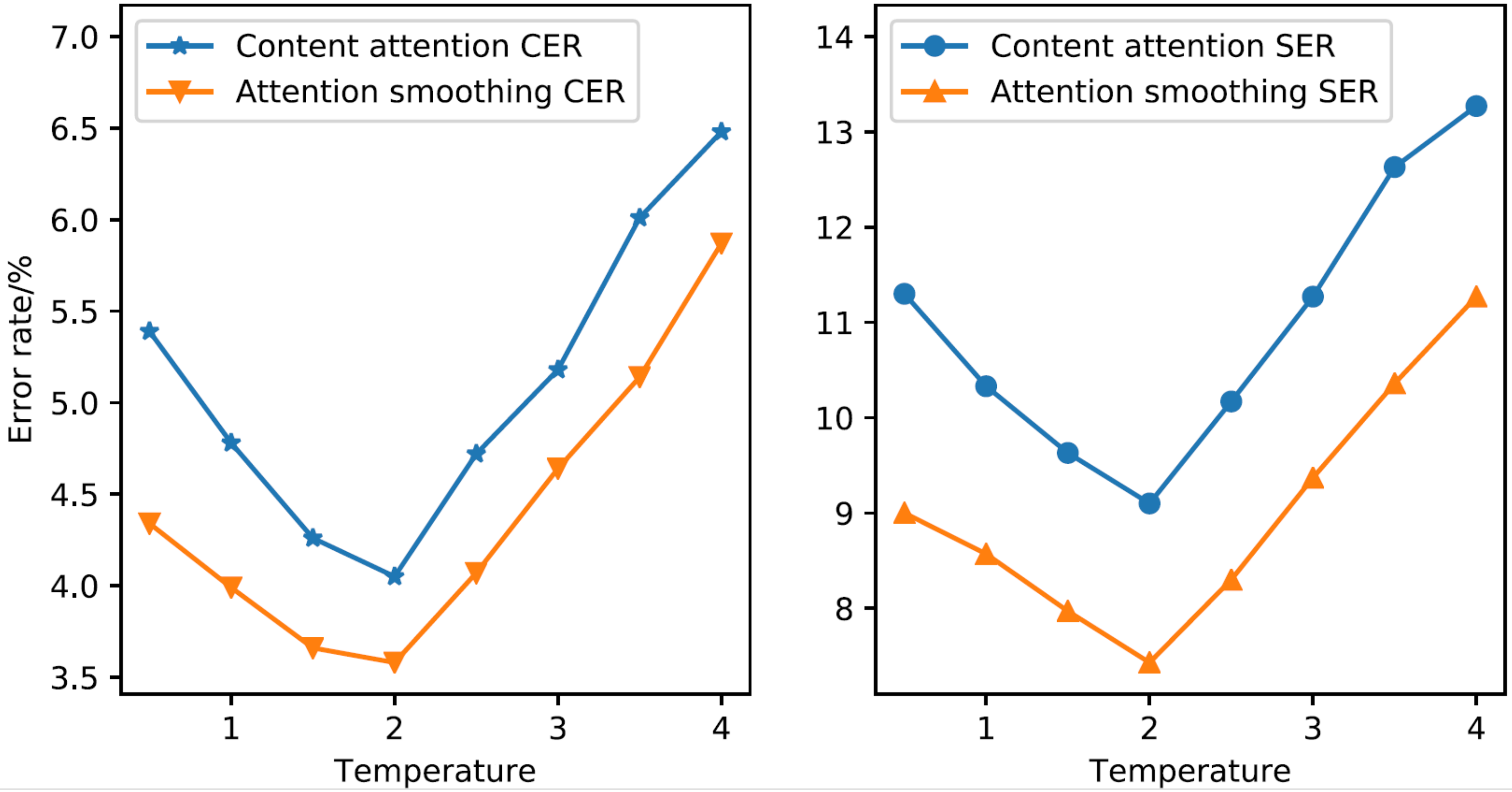
Results

- The effect of the decoding beam width for the content-based attention and attention smoothing ($\tau = 1$)



Results

- The impact of the temperature for content-based attention and attention smoothing (beam-size=30)



Results

- Results of our attention-based models with a beam size of 30, $\tau = 2$ and $\gamma = 0.1$

	CER / %	SER / %
CTC	5.29	14.57
Content based attention	4.05	9.10
+ trigram LM	3.60	7.20
Location based attention	3.82	8.17
+ trigram LM	3.26	6.33
Attention smoothing	**3.58**	**7.43**
+ trigram LM	**2.81**	**5.77**

Conclusions

- With some tricks, Mandarin LAS could be trained without pinyin
  - Embedding
  - Frame skipping
- Location-based attention is more suitable in voice search task
- Attention smoothing further improves the accuracy, and reduces the computational complexity
- An external LM can further improve the performance
- Decoding with a wider beam gives little-to-none benefit
- The temperature can smooth the distribution of characters and achieve a better result
- Our model finally achieves a CER of 3.58% and a SER of 7.43% on a Mandarin voice search task without an LM

Thanks!
Q & A