

# Recent Advances in End-to-End Speech Recognition

Shigeki Karita
NTT Communication Science Laboratories

2019/11/13, NAIST

http://karita.xyz/talk/naist2019.pdf

#### NTT C

#### **Self introduction**

- Shigeki Karita (苅田 成樹)
- NTT Communication Science Laboratories (2016-)
  - Osaka University (2010-2016)
- Research: speech processing
- Favorite programming language: dlang



- Contact
  - GitHub <a href="https://github.com/ShigekiKarita">https://github.com/ShigekiKarita</a>
  - Twitter <a href="https://twitter.com/ShigekiKarita">https://twitter.com/ShigekiKarita</a>
  - Email <u>karita@ieee.org</u>



# **Agenda**



- 1. Introduction of NTT CS Lab
- 2. Overview of End-to-End Speech Recognition
- 3. Semi-supervised End-to-End Speech Recognition
- 4. Transformer-based End-to-End Speech Recognition
- 5. Summary

Feel free to ask me anytime if you have any questions (in EN/JP)

PDF: http://karita.xyz/talk/naist2019.pdf

# **Agenda**



- Introduction of NTT CS Lab
- 2. Overview of End-to-End Speech Recognition
- 3. Semi-supervised End-to-End Speech Recognition
- 4. Transformer-based End-to-End Speech Recognition
- 5. Summary

# Feel free to ask me anytime if you have any questions (in EN/JP)

PDF: http://karita.xyz/talk/naist2019.pdf



# PART1:

# Introduction of NTT CS Lab

#### NTT (

# NTT Group and R&D

- NTT is a holding company
- NTT Group provides the largest telecommunication services.
- NTT R&D does R&D of science and technologies mainly for NTT Group

Holding company

NTT

NTT R&D

2,500

Staff
Section

#### Operation companies



Regional
Communications
Businesses



Long-Distance and International Communications Businesses

итт росомо, імс. **döcomo** 

Mobile Communications Businesses NTT DATA Corp.

NTTData

Communication
Businesses

Real Estate Financing Construction Etc.

Other Businesses

#### **NTT R&D Numbers at a Glance**



2,500

researchers working 16,000+

patents granted

\$1.0 billion invested in R&D

1,300 technical papers annually

3 times awarded IEEE Milestones

40
IEEE Fellows
(including former colleagues)



# **NTT R&D Organization**





#### President & Vice President

#### R&D Planning Department [0]

#### Service Innovation Laboratory Group [Y]

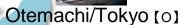
Communication services

Service Evolution Laboratories [Y·M]

Media Intelligence Laboratories [Y·M]

Software Innovation Center [M·S]

Secure Platform Laboratories [M]







Tsukuba [⊤]

Shinagawa/Tokyo [s]

#### Information Network Laboratory Group [M]

Information networks

Network Technology Laboratories [M]

Network Service Systems Laboratories [M]

Access Network Service Systems Laboratories [T-Y-M]







Atsugi [A]

Musashino/Tokyo [M]! technologies

#### Science and Core Technology Laboratory Group [A]

Cutting-edge

Network Innovation Laboratories [Y•M]

Device Innovation Center [A]

**Device Technology Laboratories** 

Communication Science Laboratories [K • A]

Basic Research Laboratories (A)



Kyoto [K]



#### NTT Innovation Institute Inc. [V]

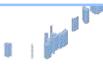
Cloud development

### **NTT Communication Science Laboratories**



### "Keihanna" (Kyoto)

- Signal processing
- Linguistic intelligence
- Learning and Intelligent Systems
- Interaction
- Machine learning







Atsugi

- Media search technology
- Human information science
- Quantum Computing
- Audio coding etc

See also: http://www.kecl.ntt.co.jp/rps/english/organogram\_e.html

#### Research Environment at NTT CS Lab



Keihanna SLURM system in 2019:

- 2460 CPU cores (INTEL Xeon)
- 412 GPUs (NVIDIA V100, 2080Ti, etc.)
- InfiniBand
- + group specific servers
- + cloud servers



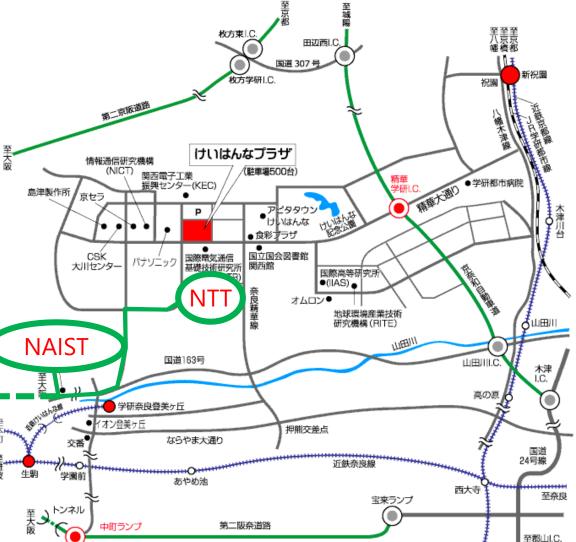


#### NTT CS Lab Keihanna ⇔ NAIST

- Very near
- Many internship students from NAIST
- Many researchers







# Signal Processing Group at NTT CS Lab



Members (+25 speech researchers in other NTT R&D)



S. Araki (leader)



M. Delcroix



N. Ito



R. Ikeshita



S. Karita



K. Arai



K. Kinoshita



T. Nakatani



T. Ochiai



A. Ogawa



N. Tawara

- More info: <a href="http://www.kecl.ntt.co.jp/media/signal.html">http://www.kecl.ntt.co.jp/media/signal.html</a>
- Open source: <a href="https://github.com/nttcslab-sp">https://github.com/nttcslab-sp</a>



# Our research goal

 Develop key technologies for understanding natural human speech interactions

Background noise

Other speakers



Reverberation

**Spontaneous speech** 

Frontend (Enhancement)

+

Backend (Recognition)

#### **Our task**



- Research interests: speech recognition/enhancement, speaker recognition/identification, language modeling, etc.
- Publish papers & file patents
  - 32 papers at ICASSP 2019
  - 18 papers at INTERSPEECH 2019
- Organize/participate challenges and conferences
  - Participant of CHiME-1, CHiME-3, REVERB
  - Organizer of REVERB, ASRU-2017, WASPAA...
- Collaboration with universities
- Occasionally, transfer our technologies to business sections

# **Agenda**



- 1. Introduction of NTT CS Lab
- 2. Overview of End-to-End Speech Recognition
- 3. Semi-supervised End-to-End Speech Recognition
- 4. Transformer-based End-to-End Speech Recognition
- 5. Summary

# Feel free to ask me anytime if you have any questions (in EN/JP)

PDF: http://karita.xyz/talk/naist2019.pdf



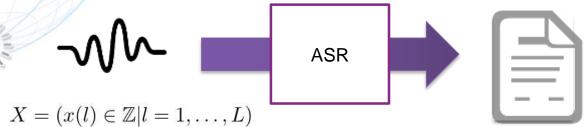
# PART2:

# Overview of End-to-End Speech Recognition

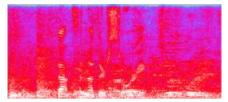


# **Automatic Speech Recognition (ASR)**

Mapping speech sequence to transcription sequence



$$X = (x(l) \in \mathbb{Z} | l = 1, \dots, L)$$
$$L = 43263$$



$$X = (\mathbf{x}_t \in \mathbb{R}^D | t = 1, \dots, T)$$
$$T = 268$$

"That's another story"

$$W = (\mathbf{w}_n \in \mathcal{V} | n = 1, \dots, N)$$
$$N = 18$$

# NTT (O)

# No official ASR example in DL frameworks?

- Tensorflow: <a href="https://github.com/tensorflow/examples">https://github.com/tensorflow/examples</a>
- → NO
- PyTorch: <a href="https://github.com/pytorch/examples">https://github.com/pytorch/examples</a>
- $\rightarrow$  NO
- Chainer: <a href="https://github.com/chainer/chainer/tree/master/examples">https://github.com/chainer/chainer/tree/master/examples</a>
- → NO
- MxNet: <a href="https://github.com/apache/incubator-mxnet/tree/master/example/speech\_recognition">https://github.com/apache/incubator-mxnet/tree/master/example/speech\_recognition</a>
- → YES!

Why ASR is not popular in NN frameworks? Let me know!

#### NTT (

# **ASR** problems

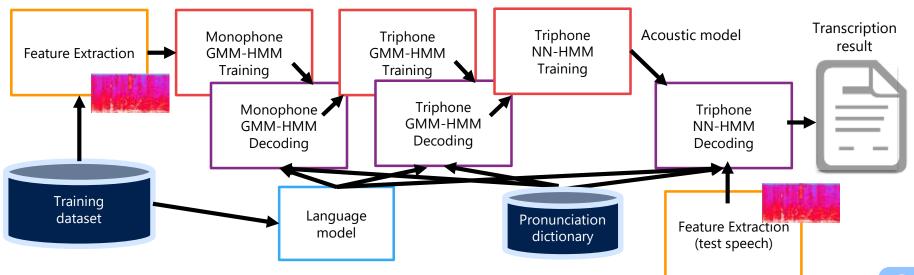
- Learning sequences
- Differences in source and target sequences
  - Length (speech is typically longer than text)
  - Modality (speech is continuous but text is discrete)



# **ASR** methods (before End-to-End)

- Learning sequences
  - → Hidden Markov model (HMM) with neural networks (NN)
- Differences in source and target sequences
  - Length (speech is typically longer than text)
  - Modality (speech is continuous but text is discrete)
  - → Acoustic and language models, pronunciation dictionary, weighted finite-state transducers (WFST)

#### The previous HMM-based ASR systems are really complicated ...





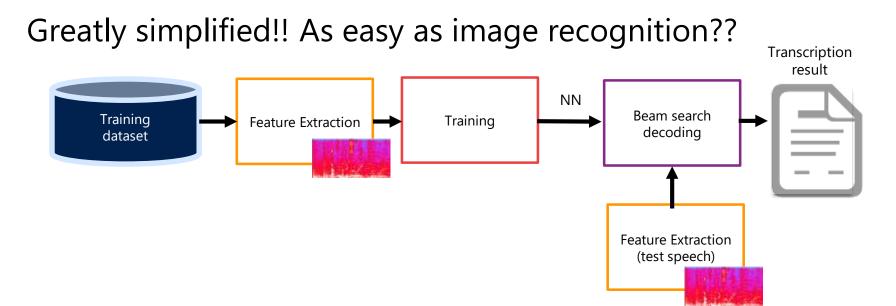
# **ASR problems (before End-to-End)**

- Learning sequences
- Differences in source and target sequences
  - Length (speech is typically longer than text)
  - Modality (speech is continuous but text is discrete)
- Complicated pipeline...



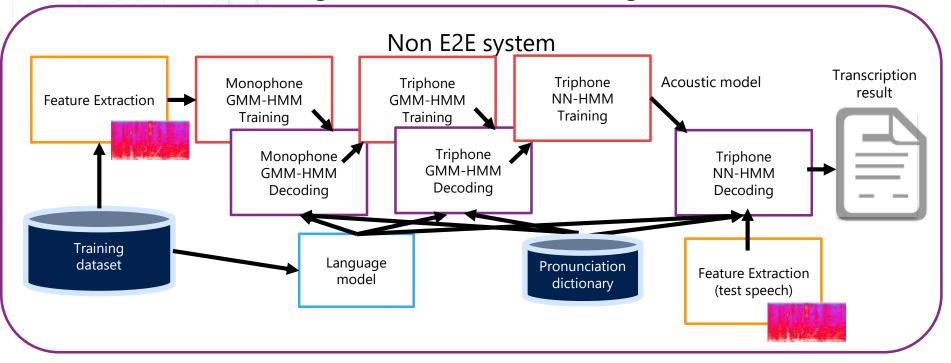
# **End-to-End (E2E) ASR methods**

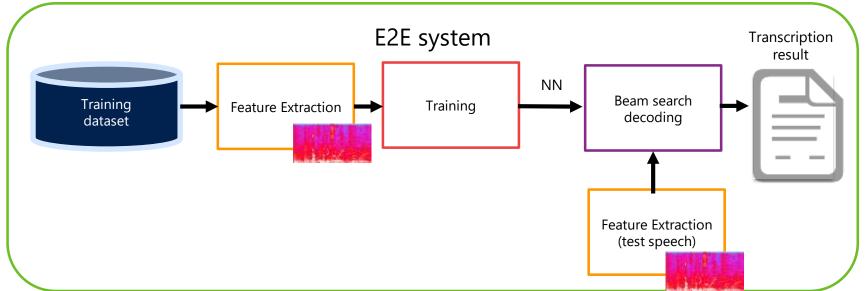
- Learning sequences
  - → Special neural networks (recurrent, convolutional, etc.)
- Differences in source and target sequences
  - Length (speech is typically longer than text)
  - Modality (speech is continuous but text is discrete)
  - → Connectionist temporal classification (CTC), sequence-to-sequence (S2S)



# Which one do you like to study?









# Which one do you like to study?

- I think this is the reason why ASR is not popular in the frameworks
- In fact, this is the E2E ASR example using CTC

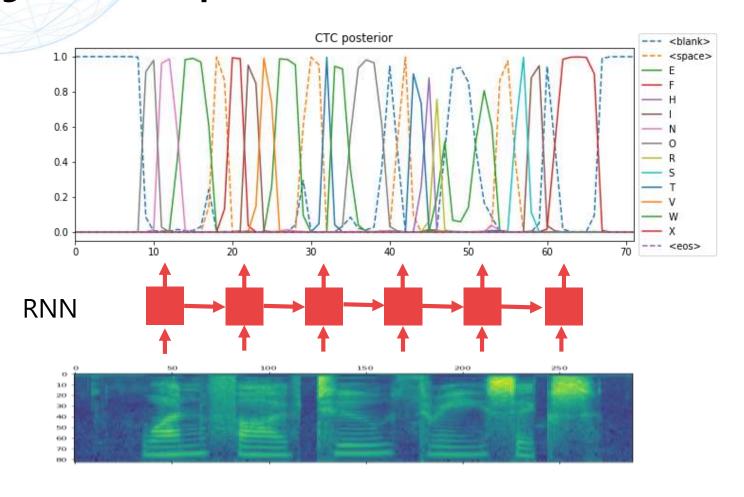
- MxNet: <a href="https://github.com/apache/incubator-mxnet/tree/master/example/speech\_recognition">https://github.com/apache/incubator-mxnet/tree/master/example/speech\_recognition</a>
- → YES!

# Connectionist temporal classification (CTC) ©

- The early E2E ASR method similar to HMM
  - A. Graves et al, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," ICML2016
- Input: a sequence of speech feature frames
- **Output**: a sequence of transcription symbol probability aligned to the speech

# Connectionist temporal classification (CTC)

- Input: a sequence of speech feature frames
- Output: a sequence of transcription symbol probability aligned to the speech

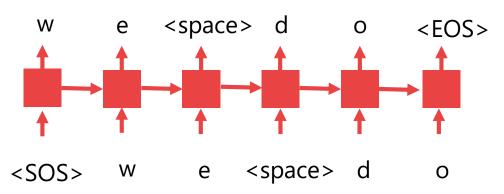




# Sequence-to-sequence (S2S)

- The later E2E ASR method similar to RNN language models (RNNLMs)
  - J. Chorowski et al., "Attention-based Models for Speech Recognition,"
     NIPS2014 Deep Learning Workshop
  - RNNLM iteratively predicts the next symbols:

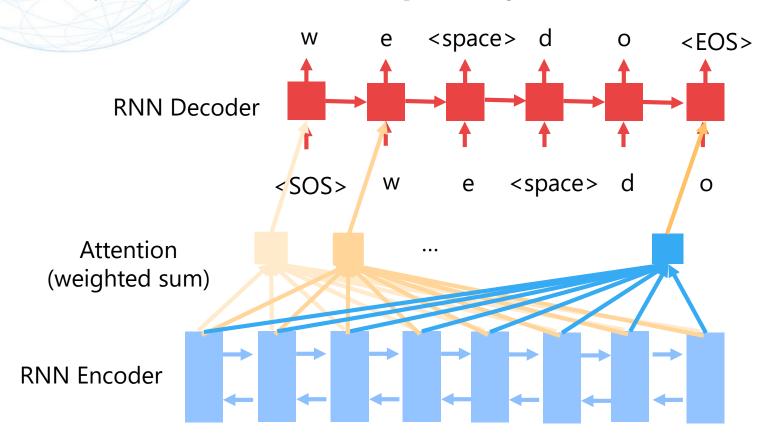






# Sequence-to-sequence (S2S)

- Input: a sequence of speech feature frames and the previous transcription symbol (and NN states)
- Output: the next transcription symbol



#### NTT (

# **Summary: PART2**

- The non-E2E ASR system got too complicated
- E2E methods (CTC and S2S) greatly simplified ASR systems
- Start your ASR/TTS research!!
- More info:
  - Takaaki Hori, Tomoki Hayashi, Shigeki Karita, Shinji Watanabe, "Advanced methods for neural end-to-end speech processing – unification, integration, and implementation," INTERSPEECH2019 <a href="https://github.com/espnet/interspeech2019-tutorial">https://github.com/espnet/interspeech2019-tutorial</a>
    - Demo with a free GPU available in Google Colab!
  - Steve Renals and Hiroshi Shimodaira, "AUTOMATIC SPEECH RECOGNITION (ASR) 2018-19: LECTURES"
    - http://www.inf.ed.ac.uk/teaching/courses/asr/lectures-2019.html
    - The greatest lecture at the university of Edinburgh!

# **Agenda**



- Introduction of NTT CS Lab
- Overview of End-to-End Speech Recognition
- Semi-supervised End-to-End Speech Recognition
- Transformer-based End-to-End Speech Recognition
- Summary

# Feel free to ask me anytime if you have any questions (in EN/JP)

PDF: http://karita.xyz/talk/naist2019.pdf



# PART3:

# Semi-supervised End-to-End Speech Recognition

#### NTT (

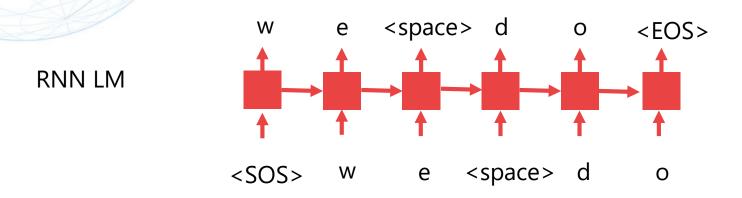
# Motivation: semi/unsupervised learning

- Paired dataset: expensive, small
- Unpaired dataset: cheap, large
  - "careful transcripts of speech average 20xRT or 20 hours for each hour of conversation ..."
  - C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a Resource for the Next Generations of Speech-to-Text," International Conference on Language Resources and Evaluation, vol. 4, pp. 69–71, 2004.



# Recap: sequence-to-sequence (S2S)

- The later E2E ASR method similar to language models (LMs)
  - RNNLM iteratively predicts the next symbols:



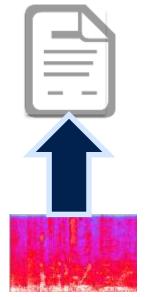
Can we use LM as unsupervised training in the ASR model?

# **Proposed method: overview**

#### NTT 😃

#### Supervised

- ASR: speech-to-text
- TTS: text-to-speech





# Proposed method: overview

#### NTT (

#### Supervised

- ASR: speech-to-text
- TTS: text-to-speech

#### Unsupervised

- Text autoencoder (TAE): text-to-text
- Speech autoencoder: (SAE): speech-to-speech

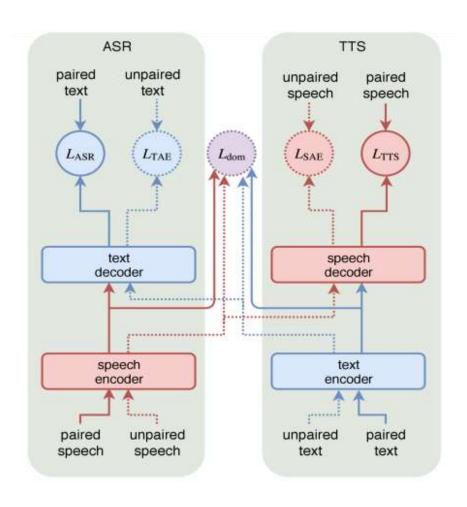
Can we build ASR/TTS using TAE and SAE?





# **Proposed method: overview**

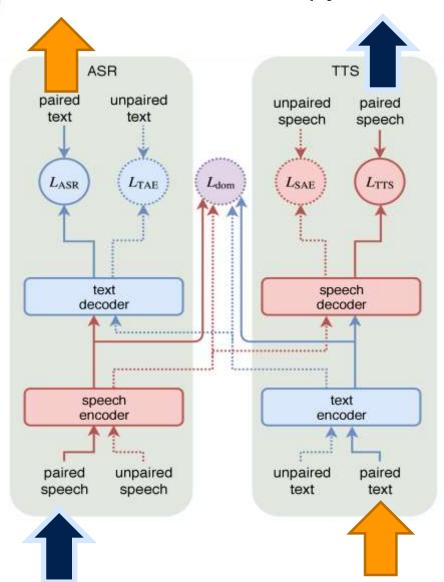
- Combine ASR and TTS as two autoencoders (speech and text)
  - with an inter-domain loss  $L_{dom}$





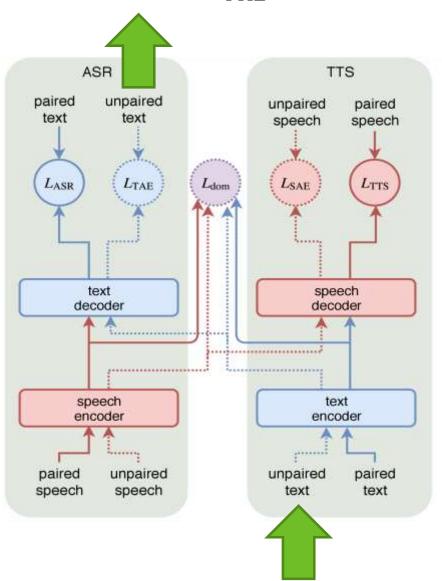
#### Proposed method: supervised training

• Conventional ASR ( $L_{ASR}$ : cross entropy) and TTS ( $L_{TTS}$ : L1+L2)



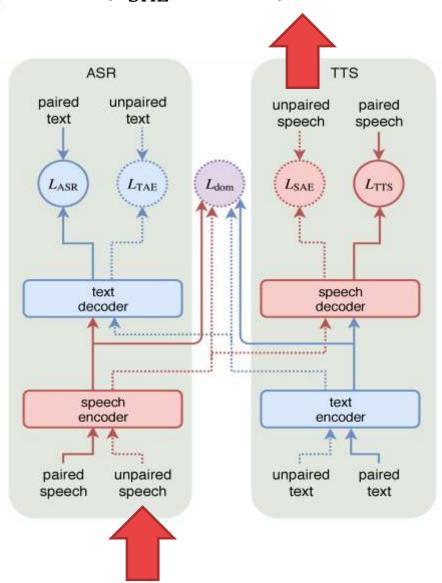
## Proposed method: unsupervised text training OFTT (2)

• Text autoencoder: usual LM ( $L_{TAE}$ : cross entropy)



# Proposed method: unsupervised speech training

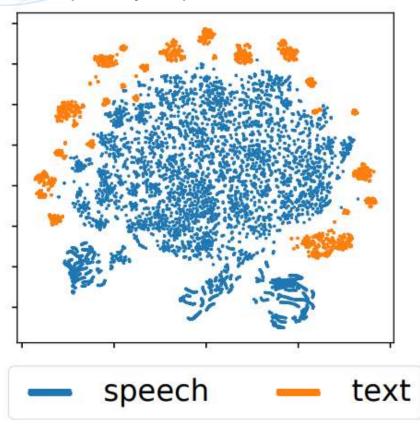
• Speech autoencoder ( $L_{SAE}$ : L1+L2)

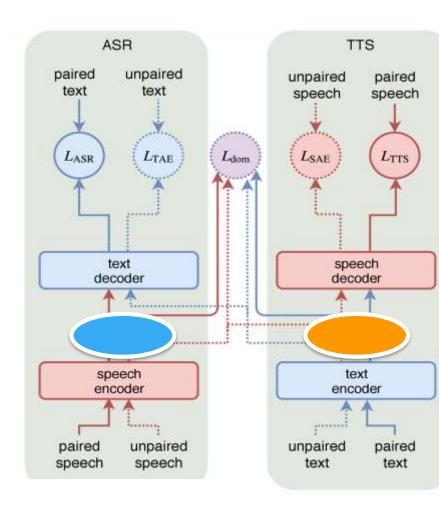


#### NTT (

#### **Problem: feature mismatch**

- Incompatible features?
- t-SNE 256D → 2D visualization
  - Completely separated...







#### Proposed method: inter-domain loss

Minimize distance between features of speech  $h^X$  and text  $h^Y$ 

- KL divergence: assume two distributions as Gaussians
- Generative Adversarial Networks: let NN learn distance between two
- Maximum Mean Discrepancy (MMD)
  - Kernel mean of minibatches of  $h^X$ ,  $h^Y$  with Gaussian kernel function k:  $E[K^{XX} 2K^{XY} + K^{YY}]$  where

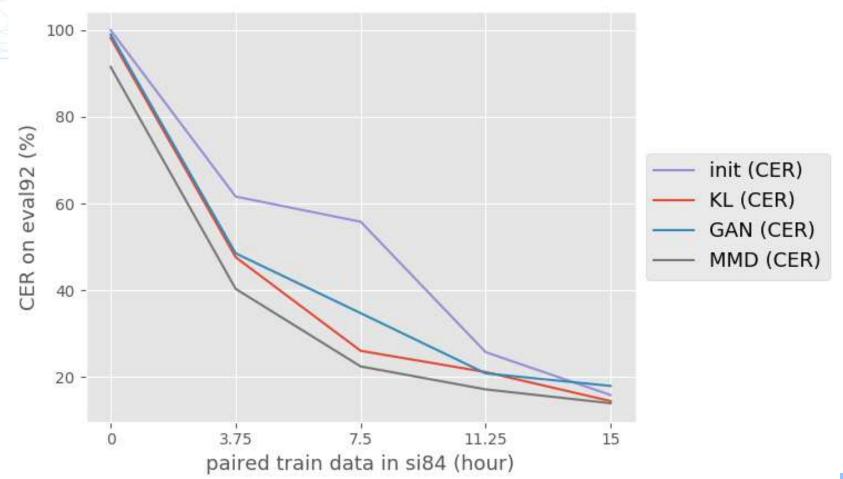
$$K^{XX} = \sum_{t=1}^{T} \sum_{t'=1}^{T} \frac{k(h_t^X, h_{t'}^X)}{T^2}, K^{YY} = \sum_{s=1}^{S} \sum_{s'=1}^{S} \frac{k(h_s^Y, h_{s'}^Y)}{S^2},$$

$$K^{XY} = \sum_{t=1}^{T} \sum_{s=1}^{S} \frac{k(h_t^X, h_s^Y)}{TS}$$

$$k(a, b) = \exp(-|a - b|^2)$$

## Preliminary experiment: dataset size and CERNTT (9)

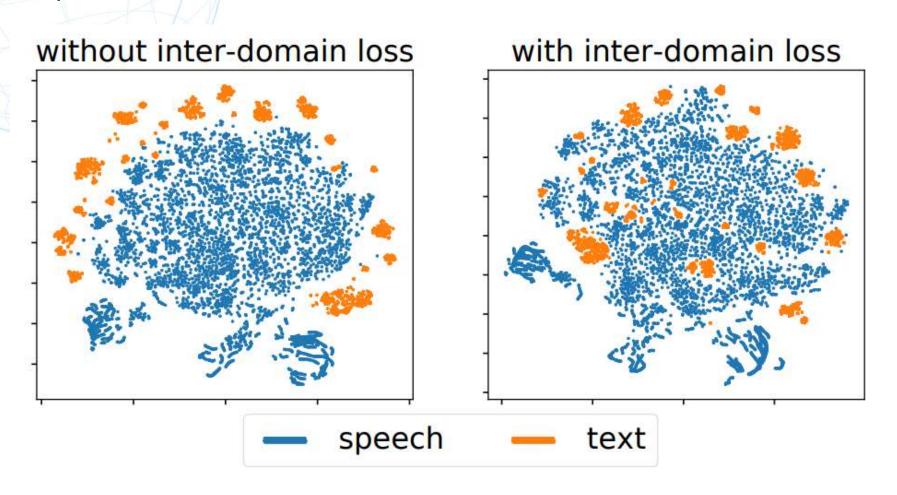
- Experiments on WSJ SI84 (paired) and SI284 (unpaired)
  - Init: initial NN trained with the small paired train dataset
  - KL/GAN/MMD: retrained NN trained with the paired and unpaired





#### **Preliminary experiment: visualization**

Experiments on WSJ SI84



#### NTT (

#### **Experiment settings**

- ASR
  - Input: 80-dim FBANK
  - Output: characters, [sos] and [eos]
  - Layers: VGG+BLSTMP enc + BLSTMP dec
- TTS
  - Input: characters + speaker embeddings [4]
  - Output: 80-dim FBANK, stop flag
  - Layers: Tacotron2 [3], same hidden dim to ASR
- Dataset: LibriSpeech official subsets (100h, 360h, 500h)
- Supervised ASR/TTS training (baseline)
  - Dataset: train clean 100 (h) for ASR (subset)
  - Adam, learning rate=1e-3, minibatch=64, 30 epochs
- Semi-supervised ASR+TTS+SAE+TAE retraining
  - Dataset: train clean 100 (h) for ASR/TTS (subset)
     train clean 360 (h) for SAE (speech only)
     train other 500 (h) for TAE (text only)
  - Adam, learning rate=1e-5, minibatch=64, 10 epochs
- LM: word-level RNNLM trained on external language resource

# **Experiment results: char error rate (CER)** NTT (S)

	multi-task loss				dev clean	test clean	
	ASR	TTS	SAE	TAE	CER/WER	CER/WER	
supervised (100h)	✓	<u>ar</u> 55	÷	-	14.1 / 26.2	15.0 / 25.0	
+ char LM	✓	-	-	-	12.2 / 22.8	11.9 / 22.5	
+ word LM	✓	=	85	-	12.1 / 21.3	11.7 / 22.0	
+ ext LM (baseline)	$\checkmark$	2	W.	-	10.3 / 20.6	10.4 / 20.6	
KL + ext LM	✓	✓	-	:	9.6 / 19.4	9.5 / 19.2	
	✓	-	V		11.6 / 21.8	12.0 / 22.1	
	✓	<u> </u>	-	✓	10.3 / 19.7	10.5 / 19.9	
	$\checkmark$	<b>√</b>	<b>V</b>	$\checkmark$	12.2 / 21.8	12.1 / 21.8	
MMD + ext LM	✓	<b>√</b>	1077	89 <del>-8</del>	9.6 / 19.1	9.5 / 18.9	
	✓	2	$\checkmark$	_	9.2 / 18.9	8.7 / 18.4	
	✓	-	-	$\checkmark$	9.4 / 18.9	9.0 / 18.4	
	✓	✓	✓	<b>√</b>	8.9 / 18.5	8.4 / 18.0	



#### **Related works**

 Copied from Murali Karthick Baskar et al., "Semi-Supervised Sequence-to-Sequence ASR Using Unpaired Speech and Text," INTERSPEECH2019 <a href="https://www.isca-speech.org/archive/Interspeech 2019/pdfs/3167.pdf">https://www.isca-speech.org/archive/Interspeech 2019/pdfs/3167.pdf</a>

	WSJ-SI84 (parallel) + WSJ-SI284 (unpaired)							
	Model	Type	RNNLM	%CER	%WER			
	Speech chain [13]	Both	=	9.9				
,	Adversarial [14]	Both	yes	=	24.9			
/	this work	Both	(2)	9.1	26.1			
	this work	Both	yes	7.8	20.3			
	oracle	25		5.5	16.4			
	oracle [29]	-	yes	2.0	4.8			
/	Librispeech 100 h (parallel) + 360 h (unpaired)							
ours	Backtranslation-TTE [10]	Text	-	10.0	22.0			
	this work	Text Text	- yes	8.0 9.1	17.9 17.3			
	Crictizing-LM [12]							
	this work	Text	yes	8.0	17.0			
\	Cycle-TTE [9]	Speech	yes	9.9	19.5			
	this work	Speech	yes	7.8	16.8			
	Adversarial-AE [15]	Both	yes	8.4	18.0			
	this work	Both		7.6	17.5			
	this work	Both	yes	7.6	16.6			
	oracle [9]	-	-	4.6	11.8			

#### NTT (

### **Summary**

- Semi-supervised ASR aims to exploit unpaired text and speech only datasets
- Our proposed ASR+TTS+SAE+TAE joint training and MMD based inter-domain loss can utilize more large unpaired datasets and result lower CER.
- Our system reduced the character error rate (CER) from 10.4% to 8.4% with the small paired data TTS joint training and large unpaired data autoencoding in the LibriSpeech small set.
- https://github.com/nttcslab-sp/espnet-semi-supervised

### **Agenda**



- 1. Introduction of NTT CS Lab
- 2. Overview of End-to-End Speech Recognition
- 3. Semi-supervised End-to-End Speech Recognition
- 4. Transformer-based End-to-End Speech Recognition
- 5. Summary

# Feel free to ask me anytime if you have any questions (in EN/JP)

PDF: http://karita.xyz/talk/naist2019.pdf



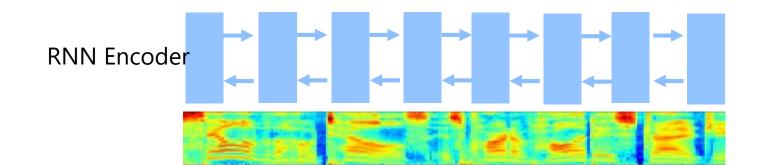
# PART4: Transformer-based End-to-End Speech Recognition

#### NTT (

- Seq2seq models directly learn speech-to-text mapping
- Encoder and decoder networks are often RNN

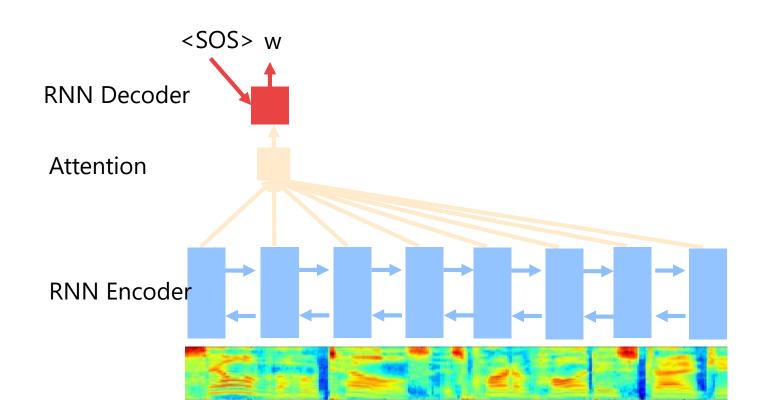


- Seq2seq models directly learn speech-to-text mapping
- Encoder and decoder networks are often RNN



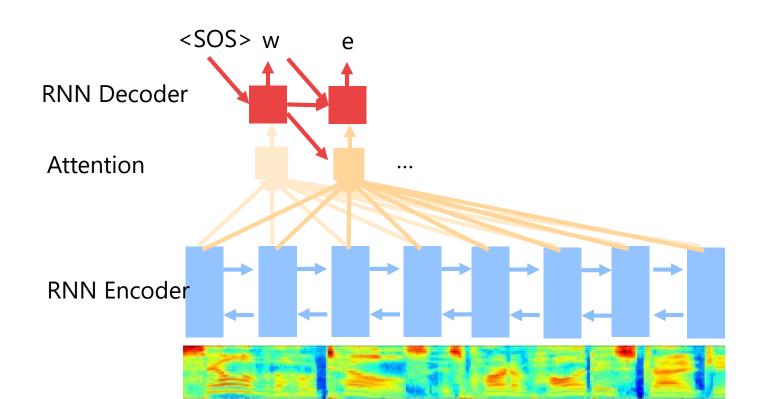


- Seq2seq models directly learn speech-to-text mapping
- Encoder and decoder networks are often RNN



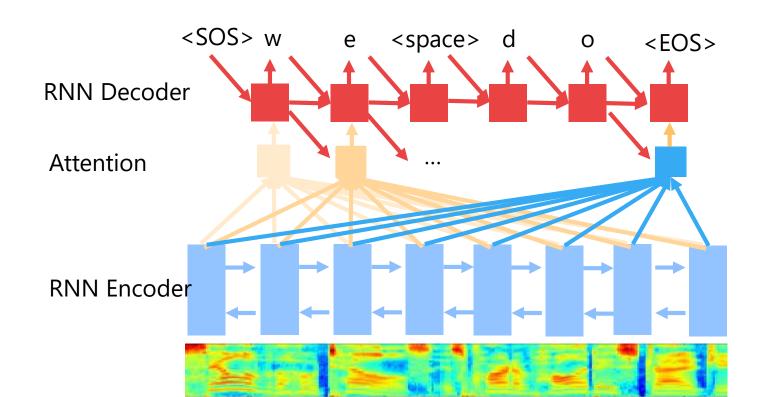


- Seq2seq models directly learn speech-to-text mapping
- Encoder and decoder networks are often RNN





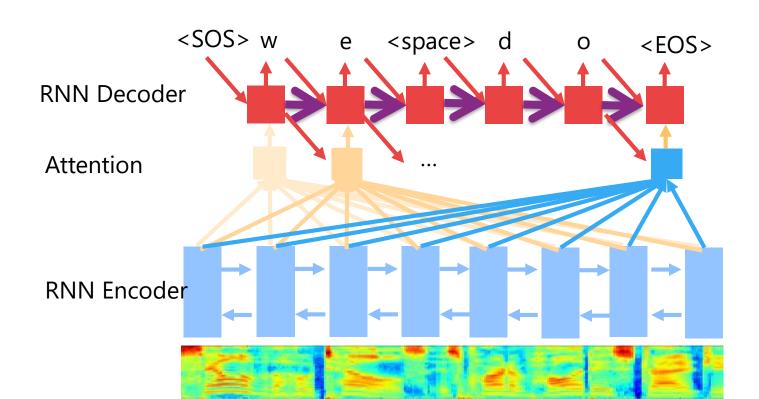
- Seq2seq models directly learn speech-to-text mapping
- Encoder and decoder networks are often RNN





- Seq2seq models directly learn speech-to-text mapping
- Encoder and decoder networks are often RNN

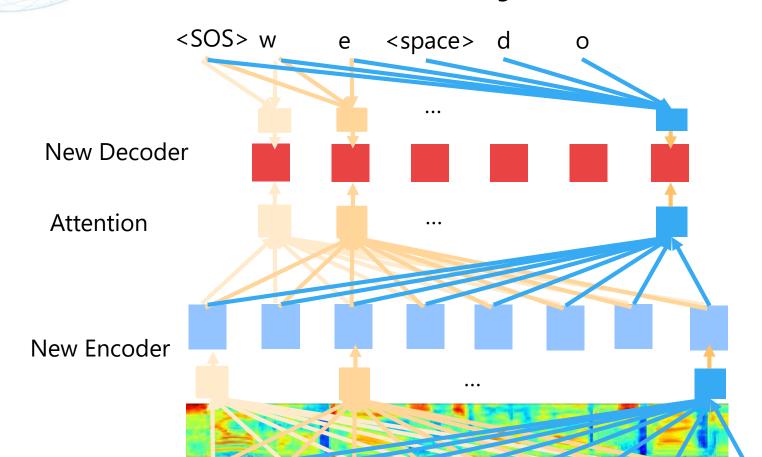
Question: Can we remove this recurrent connection?





Q: Can we remove this recurrent connection? Then, we can train NN purely in parallel

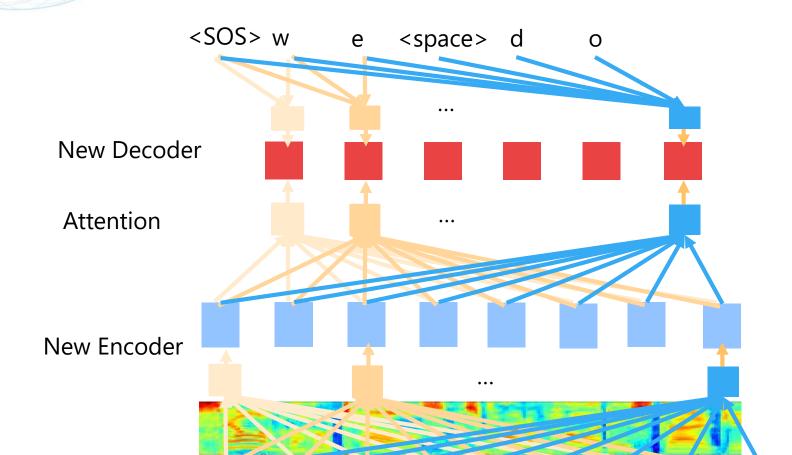
#### **Answer: use attentions everywhere**



#### **Transformer**



- Originally proposed in machine translation [Vaswani17]
- Transformer-based ASR [Dong18]
- Self attentions in encoder and decoder



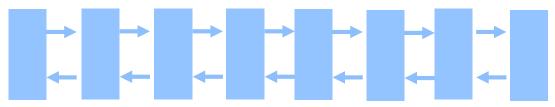


#### Transformer vs RNN encoder decoder

- RNN
  - Needs to wait for the previous output:

$$\begin{aligned} & \mathbf{h_t} = x_t W_x + \mathbf{h_{t-1}} W_h \\ & (t = 1, 2, \dots, T, x \in R^{T \times N}, h \in R^{T \times M}, W_x \in R^{N \times M}, W_h \in R^{M \times M}) \end{aligned}$$

Decoder needs to be unidirectional



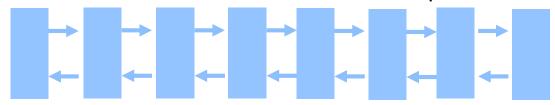
#### NTT (

#### Transformer vs RNN encoder decoder

- RNN
  - Needs to wait for the previous output:

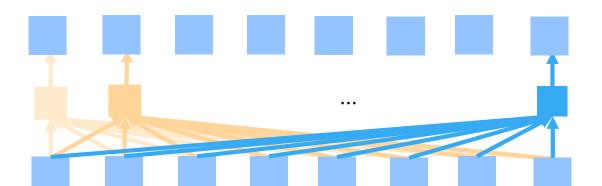
$$\begin{aligned} & \mathbf{h_t} = x_t W_x + \mathbf{h_{t-1}} W_h \\ & (t = 1, 2, \dots, T, x \in R^{T \times N}, h \in R^{T \times M}, W_x \in R^{N \times M}, W_h \in R^{M \times M}) \end{aligned}$$

Decoder needs to be unidirectional not to input future frames



- Transformer
  - Everything in self-attention is parallel:

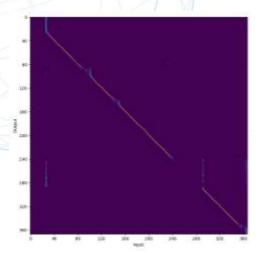
$$h = A(x)(xW_q), \qquad A(x) = \operatorname{softmax}((xW_v)(xW_k)^T)$$
$$(x \in R^{T \times N}, h \in R^{T \times M}, W_v, W_k, W_q \in R^{N \times M})$$

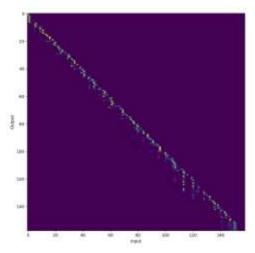


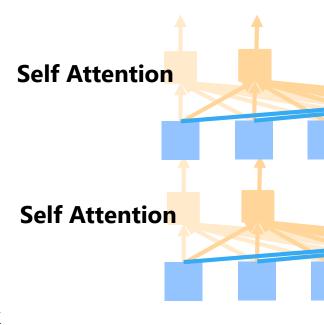
#### **Attentions in Transformer**



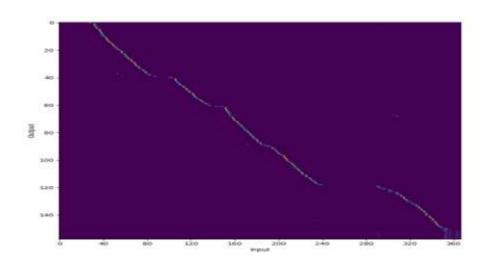
Self attention in encoder and decoder networks

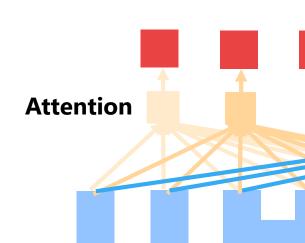






- Attention between encoder and decoder networks
  - I refer to this as "Attention" in the rest of this talk





#### **Motivation**



- ASR performance in previous work
  - RNN < Transformer [Dong18] < RNN+CTC [Watanabe18]</p>
  - Can Transformer+CTC be better?
  - Apply Transformer to more ASR tasks (not only WSJ)
- In our preliminary experiments
  - Transformer's training convergence is slower than RNN
  - Transformer results in higher word error rate (WER) than RNN+CTC in our system as well as previous work
  - Difficulty in decoding Transformer with language model (LM)

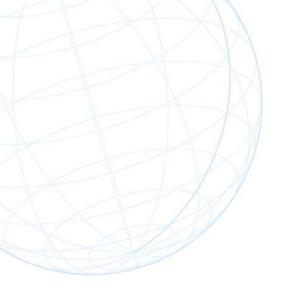
#### Contribution



#### Improve Transformer for faster training and lower WER

- Revisit effective methods in RNN [Watanabe18]
  - Connectionist temporal classification (CTC)
  - Language modeling (LM)
- Analyze settings and results in training and decoding
  - Training speed
  - Char/word error rate
  - Model size
- Open source reproducible implementation <u>https://github.com/espnet/espnet</u>



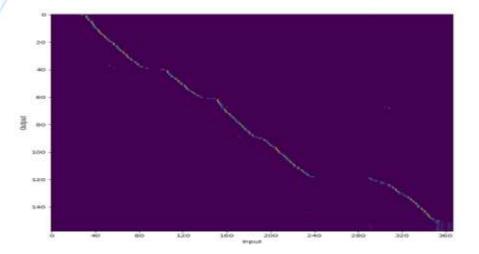


#### Proposed Method: Transformer with CTC and LM



### **Why CTC Joint Training**

- Attention plot is useful for monitoring convergence
  - It often gets monotonic in ASR

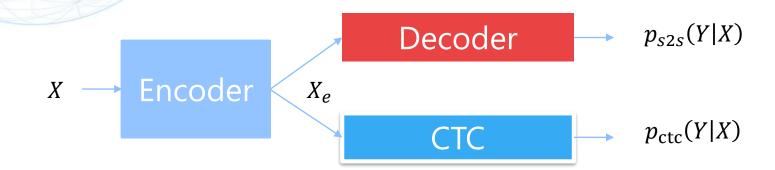


- Transformer attention is not monotonic until ~20 epochs
  - While RNN attention is monotonic at ~3 epochs
- How to accelerate monotonic attention in Transformer?
  - Window diagonal elements [Bahdanau16]: higher WER
  - Guided attention loss [Tachibana18]: higher WER
  - Connectionist temporal classification: our choice



#### **CTC Joint Training**

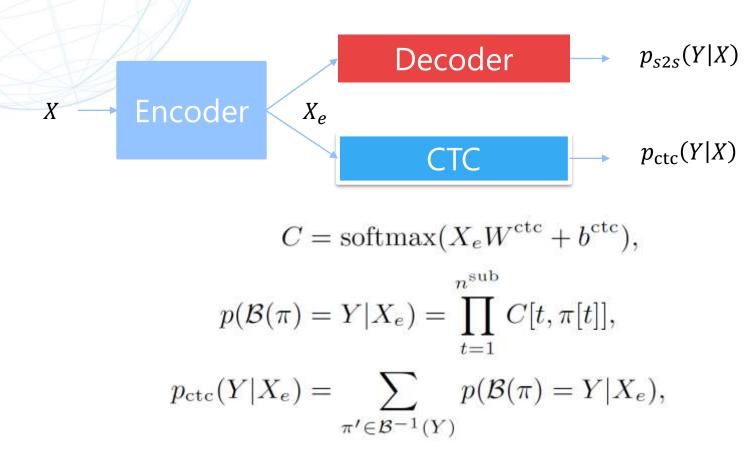
New branch from the encoder output





#### **CTC Joint Training**

New branch from the encoder output

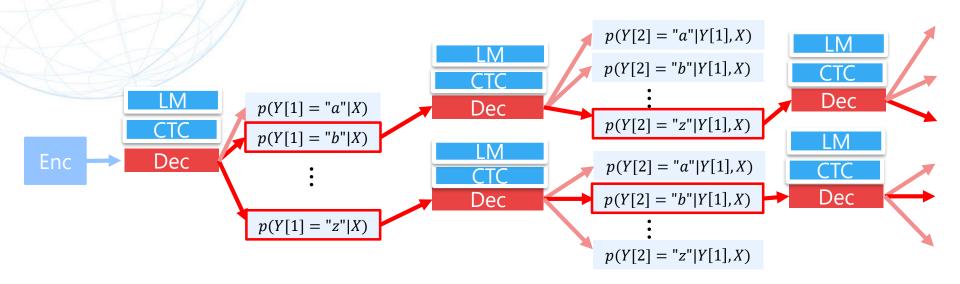


- $\pi$ : tokens at each speech frames (**alignment**)
- Objective:  $-(1 \alpha) \log p_{s2s}(Y|X) \alpha \log p_{ctc}(Y|X)$
- Explicit alignment loss makes Transformer faster and more accurate?



#### **CTC/LM Joint decoding**

Prefix tree search with the decoder, CTC and LM



Recognized text:

$$\hat{Y} = \operatorname{argmax}_{Y} \log p_{s2s}(Y|X) + \lambda \log p_{\operatorname{ctc}}(Y|X) + \gamma \log p_{\operatorname{lm}}(Y)$$





## **Experiments (INTERSPEECH 2019)**

#### **ASR NN settings**



#### RNN

- Character output
- We followed the same config in [Watanabe18]:
  - Subsampling two conv + maxpooling layers for speech input
  - 320 or 1024 units per layer (depends on tasks)
  - 6 or 8 BLSTM encoder layers, 1 or 2 LSTM decoder layer (depends on tasks)
  - Location-based attention
  - CTC joint training, CTC+RNNLM joint decoding
  - Adadelta optimizer + dev-set eps decay

#### Transformer

- Character output
- We followed the same "BIG" config in [Dong18]:
  - Subsampling two conv layers for speech input
  - 2048 feedforward units
  - 256 attention units
  - 4 heads attention
  - 12 encoder layers, 6 decoder layers
  - CTC joint training, CTC+RNNLM joint decoding
  - Adam optimizer + linear-warmup/inverse-sqrt LR scheduler

Settings except for ASR NN settings above are shared (best for the RNN system)



#### **Learning Curve**

- CTC joint training accelerated Transformer's convergence
  - Wall clock time to reach 90% valid accuracy on WSJ
    - RNN+CTC: 1.2 hour
    - Transformer: 1.8 hour
    - Transformer+CTC: 1.0 hour

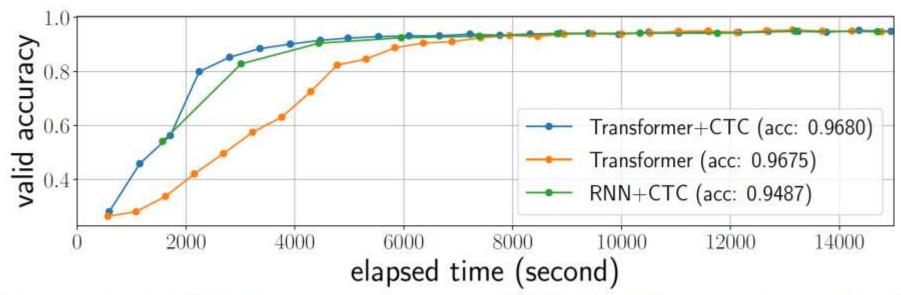


Figure 1: Validation accuracy on WSJ dev93 over time. Each data point corresponds to the end of each epoch.

#### **Attention Plot**



Transformer with CTC learns monotonic attention at 3<sup>rd</sup> epoch

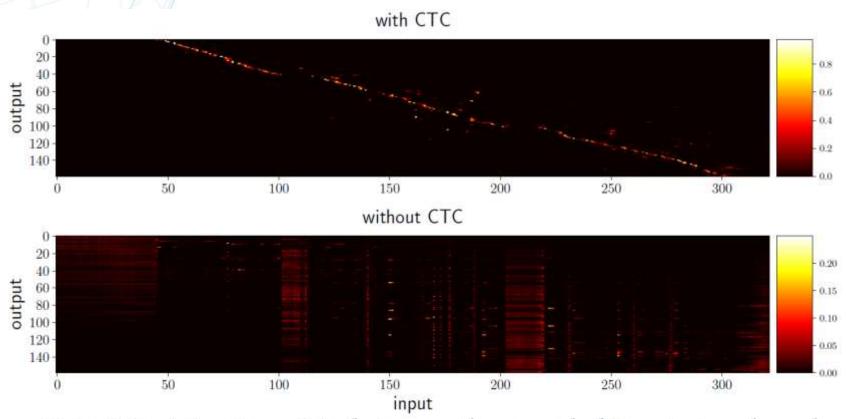


Figure 2: Attention plots between the encoded input speech and last decoder output for a WSJ 4k0c301 provided by Transformer

#### NTT (

#### **Joint Decoding**

- CTC helps joint decoding with LM
- Prefix search:

$$\hat{Y} = \operatorname{argmax}_{Y} \log p_{s2s}(Y|X) + \lambda \log p_{\operatorname{ctc}}(Y|X) + \gamma \log p_{\operatorname{lm}}(Y)$$

Table 4: Transformer WERs on WSJ dev93.

				LM $\gamma$		
		0.0	0.01	0.1	1.0	10.0
CTC $\lambda$	0.0	14.2	69.9	83.4	93.3	95.2
	0.1	13.7	13.6	12.3	7.8	86.3
	0.3	14.4	14.2	12.8	7.7	74.0
	0.5	15.7	15.5	83.4 12.3 12.8 13.5	8.5	56.9

#### **Word error rate on WSJ**

- Read English ASR task
- Transformer+CTC+LM joint decoding is the best
  - Even only CTC joint training or decoding improved Transformer

	Dev WER	Test WER
RNN	20.8	16.7
RNN + CTC/LM decode	9.0	6.0
Transformer	14.3	11.1
Transformer + CTC train	14.2	10.6
Transformer + CTC decode	13.8	10.2
Transformer + CTC/LM decode	7.7	4.5

#### **Word error rate on TED-LIUM2**

- Lecture English ASR task
- Transformer+CTC+LM joint decoding is the best
  - Even only CTC joint training or decoding improved Transformer

	Dev WER	Test WER
RNN	24.8	19.0
RNN + CTC/LM decode	19.8	18.6
Transformer	24.6	16.1
Transformer + CTC train	20.9	15.1
Transformer + CTC decode	15.8	14.2
Transformer + CTC/LM decode	13.1	11.6

#### Char/word error rates in CSJ

- Lecture Japanese ASR task
- Transformer+CTC+LM joint decoding is the best
  - Even only CTC joint training or decoding improved Transformer

	Dev CER	Eval1/2/3 CER
RNN	-	11.4 / 7.9 / 9.0
RNN + CTC/LM decode	-	6.8 / 4.8 / 5.0
Transformer	6.0	7.7 / 5.8 / 5.9
Transformer + CTC train	5.9	7.3 / 5.0 / 5.8
Transformer + CTC decode	5.8	7.1 / 4.9 / 5.6
Transformer + CTC/LM decode	5.5	5.7 / 4.1 / 4.5

We split the first 4k utterances in the official training as "dev" here. We retrieved RNN results from egs/csj/asr1/RESULTS in ESPnet LM in this result was updated after the Interspeech submission

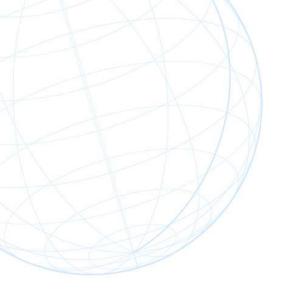
### Model parameter size

On CSJ, Transformer has the smallest size and the lowest CER.

	Model size	CSJ CER (eval1/2/3)
Transformer	120MB	5.7 / 4.1 / 4.5
RNN	476MB	6.6 / 4.8 / 5.0
Kaldi TDNN + HCLG.fst(*)	235MB	7.5 / 6.3 / 6.9

<sup>(\*)</sup> Kaldi default LF-MMI recipe at the commit "c7876a33."





# **LARGER Experiments (ASRU2019)**



#### Difference from INTERSPEECH2019

- Data augmentation
  - Speed perturbation [Ko2015]
  - SpecAugment [Park2019]
- More faster training
  - Multi GPUs (up to 8)
  - Mixed precision (fp16 + fp32)
- More experiments
  - Comparison to non-end-to-end ASR system: Kaldi
  - Total 15 monolingual + 1 multilingual ASR tasks!
    - In the paper, we also provided text-to-speech and speech translation results but I skip this today...



## **Monolingual ASR Datasets**

**Table 1**. ASR dataset description. Names listed in "test sets" correspond to ASR results in Table 2. We enlarged corpora marked with (\*) by the external WSJ train\_si284 dataset (81 hours).

dataset	language	hours	speech	test sets
AISHELL [27]	zh	170	read	dev / test
AURORA4 [28] (*)	en	15	noisy read	(dev_0330) A / B / C / D
CSJ [29]	en ja	581	spontaneous	eval1 / eval2 / eval3
CHiME4 [30] (*)	en	108	noisy far-field multi-ch read	dt05_simu / dt05_real / et05_simu / et05_real
CHiME5 [31]	en	40	noisy far-field multi-ch conversational	dev_worn / kinect
Fisher-CALLHOME Spanish	es	170	telephone conversational	dev / dev2 / test / devtest / evItest
HKUST [32]	zh	200	telephone conversational	dev
JSUT [24]	ja	10	read	(our split)
LibriSpeech [33]	en	960	clean/noisy read	dev_clean / dev_other / test_clean / test_other
REVERB [34] (*)	en	124	far-field multi-ch read	et_near / et_far
SWITCHBOARD [35]	en	260	telephone conversational	eval2000 / RT'03
TED-LIUM2 [36]	en	118	spontaneous	dev / test
TED-LIUM3 [37]	en	452	spontaneous	dev / test
VoxForge [38]	it	16	read	(our split)
WSJ [39]	en	81	read	dev93 / eval92

We tested various ASR tasks in language, hours, and recoding.



# **Monolingual ASR Results**

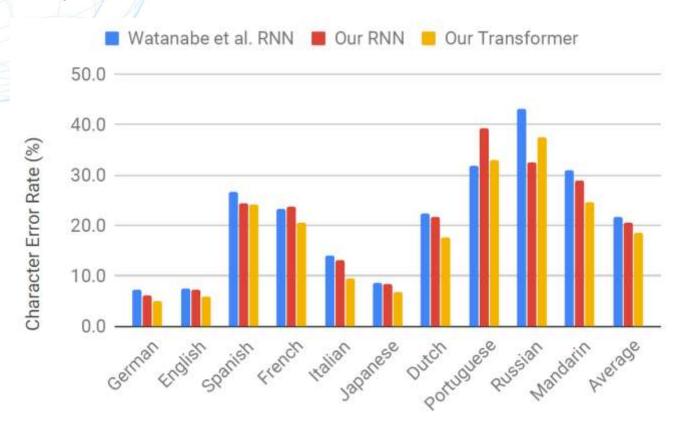
- Transformer achieved lower CER/WER than RNN in 13/15 tasks
  - Including noisy, far-field low resource, etc.

dataset	token	error	Kaldi	Our RNN	Our Transformer
AISHELL	char	CER	N/A / 7.4	6.8 / 8.0	6.0 / 6.7
AURORA4	char	WER	(*) 3.6 / 7.7 / 10.0 / 22.3	3.5 / 6.4 / 5.1 / 12.3	3.3 / 6.0 / 4.5 / 10.6
CSJ	char	CER	(*) 7.5 / 6.3 / 6.9	6.6 / 4.8 / 5.0	5.7 / 4.1 / 4.5
CHiME4	char	WER	6.8 / 5.6 / 12.1 / 11.4	9.5 / 8.9 / 18.3 / 16.6	9.6 / 8.2 / 15.7 / 14.5
CHiME5	char	WER	47.9 / 81.3	59.3 / 88.1	60.2 / 87.1
Fisher-CALLHOME Spanish	char	WER	N/A	27.9 / 27.8 / 25.4 / 47.2 / 47.9	27.0 / 26.3 / 24.4 / 45.3 / 46.2
HKUST	char	CER	23.7	27.4	23.5
JSUT	char	CER	N/A	20.6	18.7
LibriSpeech	BPE	WER	3.9 / 10.4 / 4.3 / 10.8	3.1 / 9.9 / 3.3 / 10.8	2.2 / 5.6 / 2.6 / 5.7
REVERB	char	WER	18.2 / 19.9	24.1 / 27.2	15.5 / 19.0
SWITCHBOARD	BPE	WER	18.1 / 8.8	28.5 / 15.6	26.0 / 14.0
TED-LIUM2	BPE	WER	9.0 / 9.0	11.2 / 11.0	9.3 / 8.1
TED-LIUM3	BPE	WER	6.2 / 6.8	14.3 / 15.0	9.7 / 8.0
VoxForge	char	CER	N/A	12.9 / 12.6	9.4 / 9.1
WSJ	char	WER	4.3 / 2.3	7.0 / 4.7	6.8 / 4.4



# **Multilingual ASR results**

- Previous system: [Watanabe2017] RNN
  - Except for Russian, Transformer results in the best





## **Multi GPU Training**

- Transformer could be trained faster in multi GPU
  - More #GPUs results in better accuracy
  - Scaling ASR!!

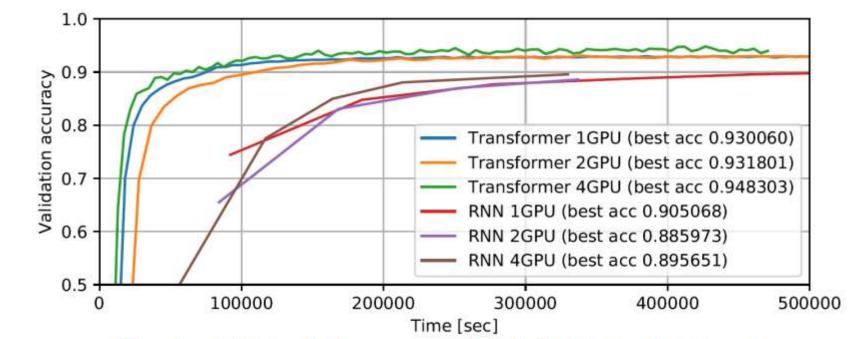
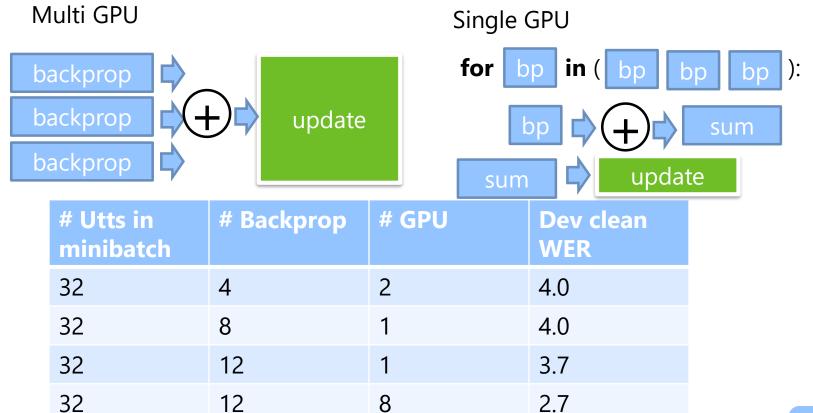


Fig. 2. ASR training curve with LibriSpeech dataset.



# **Training tips: minibatch**

- Transformer prefers larger mini batch (backprop in parallel)
  - More suitable for HPC servers
- Multiple backprop per single update (backprop interatively)
  - If you do not have HPC servers





## **Training tips: data augmentation**

- Speed perturbation (sp) [Ko2015]: speed up/down playback speed
- SpecAugment (sa) [Park2019]: random freq/time mask on spectrogram

greatly improve both Transformer and RNN

	TEDLIUM2 Dev WER	TEDLIUM2 Test WER
RNN	19.6	18.6
Transformer	12.8	11.0
RNN + sp	11.2	11.0
Transformer + sp	11.2	10.4
RNN + sa	11.4	10.6
Transformer + sa	11.2	9.6
RNN + sp + sa	11.4	10.6
Transformer + sp + sa	9.3	8.1

# **Summary**

- Transformer can replace RNN in ASR when we combined it with CTC and LM
- Our open source implementation and pretrained models: <a href="https://github.com/espnet/espnet">https://github.com/espnet/espnet</a>
- Open paper access
  - INTERPSEECH2019:

https://www.iscaspeech.org/archive/Interspeech\_2019/abstracts/1938.html

ASRU2019 preprint: <a href="https://arxiv.org/abs/1909.06317">https://arxiv.org/abs/1909.06317</a>

# **Agenda**



- 1. Introduction of NTT CS Lab
- 2. Overview of End-to-End Speech Recognition
- 3. Semi-supervised End-to-End Speech Recognition
- 4. Transformer-based End-to-End Speech Recognition
- 5. Summary

# Feel free to ask me anytime if you have any questions (in EN/JP)

PDF: http://karita.xyz/talk/naist2019.pdf

# **Summary**

- PART1: NTT is the best place for research in Japan
- PART2: E2E systems greatly simplify the ASR pipeline
  - with NN, CTC, seq2seq, etc
  - without pronunciation dictionary, multi-stage training
  - ASR research for everyone! <a href="https://github.com/espnet/espnet">https://github.com/espnet/espnet</a>
- PART3: Our semi-supervised training improves E2E ASR in lowresource scenario with unsupervised data
  - using speech/text autoencoders and inter-domain loss
- PART4: Transformer makes E2E ASR faster and more robust
  - can easily replace RNNs in many ASR tasks
- However, E2E ASR still has many unresolved problems
  - unsupervised/few-shot learning, online/real-time decoding, sequential training, joint speech enhancement, data cleaning, etc

#### **References in PART4**

- L. Dong, S. Xu, B Xu, "Speech-Transformer: A No-Recurrence Sequenceto-Sequence Model For Speech Recognition," ICASSP 2018
- Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, Tsubasa Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," INTERSPEECH 2018
- D. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. Cubuk, Q. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," INTERSPEECH 2019
- S.Watanabe, T. Hori, and J. R. Hershey, "Language independent end-toend architecture for joint language identification and speech recognition," ASRU 2017
- T. Ko, V. Peddinti, D Povey, S. Khudanpur, "Audio Augmentation for Speech Recognition," INTERSPEECH 2015
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser,
   I. Polosukhin, "Attention is All you Need," NeurIPS 2017
- H. Tachibana, K. Uenoyama, S. Aihara, "Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention," ICASSP 2018