INTERSPEECH 2019 TUTORIAL

Advanced Methods for Neural End-to-End Speech Processing

– Unification, Integration, and Implementation –

Takaaki Hori
Mitsubishi Electric Research Laboratories (MERL)

Shigeki Karita
NTT Communication Science Laboratories

Tomoki Hayashi
Nagoya University

Shinji Watanabe
Johns Hopkins University

https://github.com/espnet/interspeech2019-tutorial

ESPnet

Abbreviations

- ASR: Automatic speech recognition
- CBHG: 1-D convolution bank + highway network
- + bidirectional GRU
- CNN: Convolutional neural network
- CTC: Connectionist temporal classification
- DM: Dialog management
- DNN: Deep neural network
- E2E: End to end
- GRU: Gated recurrent unit
- HMM: Hidden Markov model
- IBM: Ideal binary mask
- Joint C/A: Joint CTC/Attention
- KL: Kullback Leibler
- LSTM: Long short-term memory
- MMD: Maximum mean discrepancy
- MSE: Mean square error
- MT: Machine translation
- MVDR: Minimum variance distortionless response
- NLU: Natural language understanding
- NLG: Natural language generation
- NMT: Neural Machine translation
- OCR: Optical character recognition
- PSD: Power spectrum density
- RNN: Recurrent neural network
- RNN-T: RNN transducer
- SE: Speech enhancement
- seq2seq: Sequence to sequence
- SLU: Spoken language understanding
- SS: Speech separation
- TTE: Text to encoder states
- TTS: Text to speech
- WPE: Weighted prediction error

Tutorial speakers

Takaaki Hori

- Senior Principal Research Scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA.
- Applications of end-to-end neural network approaches to spoken language processing including speech recognition, spoken language understanding and dialog systems.
- Led "Multilingual End-to-end ASR for Incomplete Data" research group at 2018 Jelinek Summer Workshop on Speech and Language Technology.
- Core developers of ESPnet end-to-end speech processing toolkit.

Tutorial speakers

Tomoki Hayashi

- Postdoctoral researcher @ Nagoya University
  COO @ Human Dataware Lab. Co., Ltd.
- Statistical speech and audio signal processing including speech recognition, speech synthesis, and sound event detection
- One of the organizers of the tutorial “Statistical voice conversion with direct waveform modeling” in Interspeech 2019
- Core developer of the ESPnet end-to-end speech processing toolkit, mainly developing of TTS modules in the ESPnet.

Tutorial speakers

Shigeki Karita

- Research Scientist, NTT Communication Science Laboratories, Japan
- Automatic speech recognition (semi-supervised, noise robust, far-field, sequence discriminative training, etc), and speech enhancement
- Core developer of the ESPnet end-to-end speech processing toolkit, mainly developing of ASR modules in the ESPnet.

Tutorial speakers

Shinji Watanabe

- Associate research professor at Johns Hopkins University
- Research and development of end-to-end neural network approaches applied to wide range of speech processing including speech recognition, separation, enhancement, and synthesis.
- Co-led "Multilingual End-to-end ASR for Incomplete Data" research group at 2018 Jelinek Summer Workshop on Speech and Language Technology.
- The original author of the ESPnet end-to-end speech processing toolkit.

1. Introduction to End-to-End Speech Processing

Shinji Watanabe
Johns Hopkins University

Table of contents

Table of contents

Shinji Watanabe

Table of contents

Takaaki Hori

Table of contents

1. Introduction to End-to-End Speech Processing
2. End-to-End Integration of Multiple Speech Applications
3. End-to-End Speech Processing Toolkit (ESPnet)

Break

4. Building End-to-End ASR and TTS Systems
   - TTS systems
   - ASR systems
5. Conclusion and Future Research Directions

Please access this URL and prepare for the hands-on tutorial

https://github.com/espnet/interspeech2019-tutorial

Table of contents

Tomoki
Hayashi

Table of contents

Shigeki Karita

Table of contents

High-level introduction

- **Unified** views of multiple speech processing applications based on end-to-end neural architecture
- **Integration** of these applications in a single network
- **Implementation** of such applications and their integrations based on an open source toolkit ESPnet in an unified manner

Aim of this tutorial

Easily perform speech processing research and development

- Provides a lot of case studies
- Hands-on tutorials

Target of this tutorial

- Beginner for speech processing
- Expert for one or more speech processing applications and want to extend your research topics
  - E.g., ASR expert but not for TTS and want to start the TTS study

You could learn

1. Introduction to End-to-End Speech Processing

⚒ Overviews of end-to-end speech processing (mainly ASR)

1. End-to-End Integration of Multiple Speech Applications

⚒ How to integrate these methods into a single network

1. End-to-End Speech Processing Toolkit (ESPnet)

4. Building End-to-End ASR and TTS Systems

   TTS systems

   ASR systems

⚒ Build state-of-the-art speech recognition & synthesis from scratch

Table of contents

1-a) Target applications

This tutorial covers the following applications:

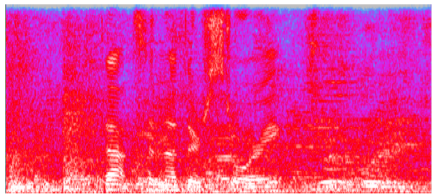
- Automatic speech recognition (ASR), Speech to text
- Speech synthesis, Text to speech (TTS)
- Speech translation (ST)
- Spoken language understanding (SLU)
- Speech enhancement (SE)

Automatic speech recognition (ASR)

- Mapping **speech** sequence to ***character*** sequence

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

$$L = 43263$$



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

$$T = 268$$

"That's another story"

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

$$N = 18$$

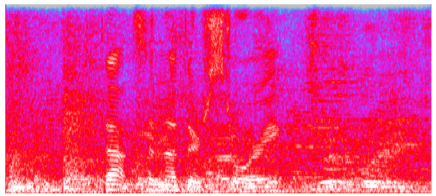
Speech to text translation (ST)

- Mapping **speech** sequence in a **source** language to ***character*** sequence in a **target** language

That's another story

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

$$L = 43263$$



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

$$T = 268$$

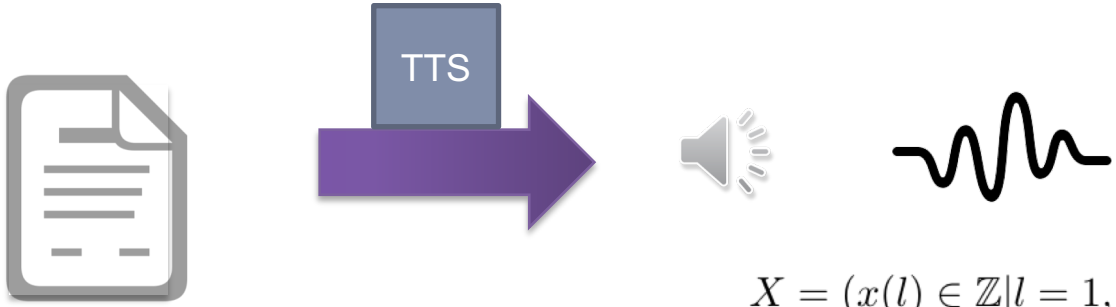
"Das ist eine andere Geschichte"

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

N=31

Text to speech (TTS)

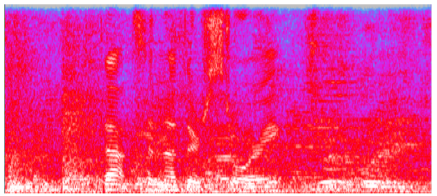
- Mapping **character** sequence to **speech** sequence



"That's another story"

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

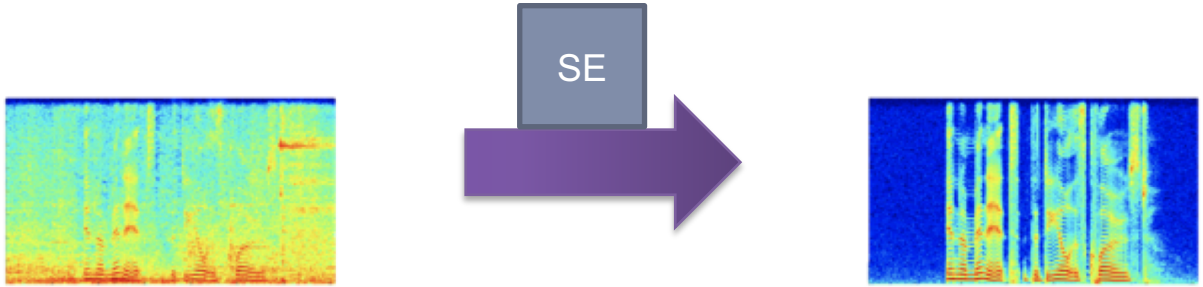
$$X = (x(l) \in \mathbb{Z} | l = 1, \ldots, L)$$
$$L = 43263$$



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

Speech enhancement (SE)

- Mapping **noisy** speech sequence to **clean** speech sequence



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

$$X' = (\mathbf{x}'_t \in \mathbb{R}^D | t = 1, \ldots, T)$$
$$T = 268$$

All of the problems

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

Unified view with sequence to sequence

- All of the above problems: find a mapping function from *sequence* to *sequence* (**unification**)

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

  - ASR: X = Speech, Y = Text
  - TTS: X = Text, Y = Speech
  - ST: X = Speech (EN), Y = Text (JP)
  - Speech Enhancement: X = Noisy speech, Y = Clean speech
- Mapping function $f(\cdot)$
  - Sequence to sequence (seq2seq) function
  - ASR as an example

Table of contents

Seq2seq end-to-end ASR

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

Mapping seq2seq function $f(\cdot)$

1. Connectionist temporal classification (CTC)
2. Attention-based encoder decoder
3. Joint CTC/attention (Joint C/A)
4. RNN transducer (RNN-T)
5. Transformer

Connectionist temporal classification (CTC)

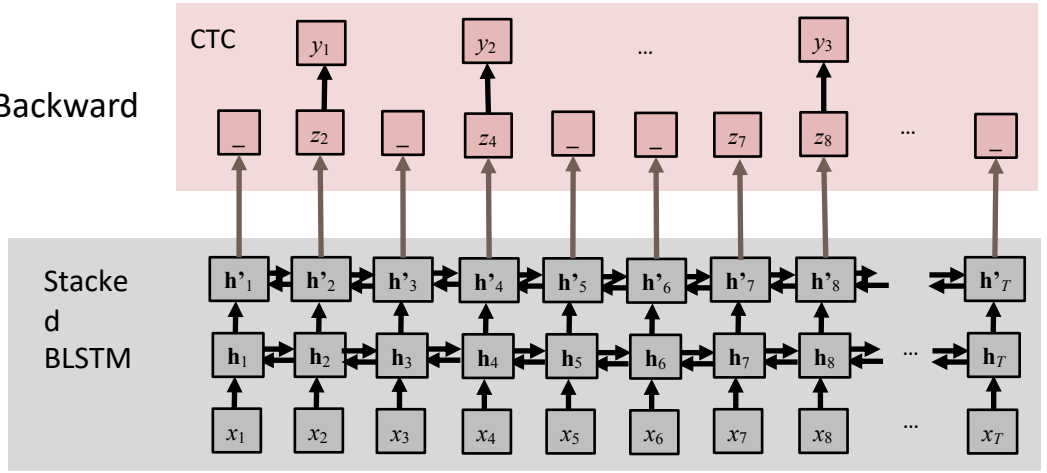
[Graves+ 2006, Graves+ 2014, Miao+ 2015]

- Use bidirectional RNNs to predict frame-based labels including blanks
- Find alignments between X and Y using dynamic programming

😄 Simple implementation (built-in & cudnn), on-line, fast

😓 Poor performance (conditional independence assumptions), limited applicationsc

Forward-Backward algorithm

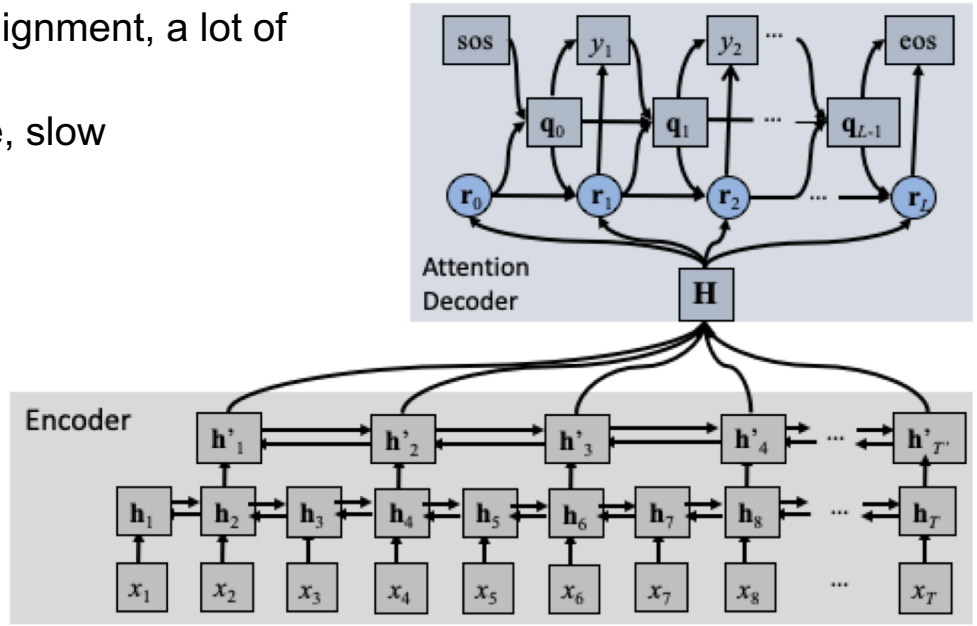


Attention-based encoder decoder [Chorowski+ 2015, Chan+ 2016]

- Encoder: acoustic model, decoder: RNN language model, attention: align input and output labels
- No conditional independence assumption

😄 Good performance but too flexible alignment, a lot of applications (ASR, TTS, NMT)

😓 Complicated implementation, off-line, slow

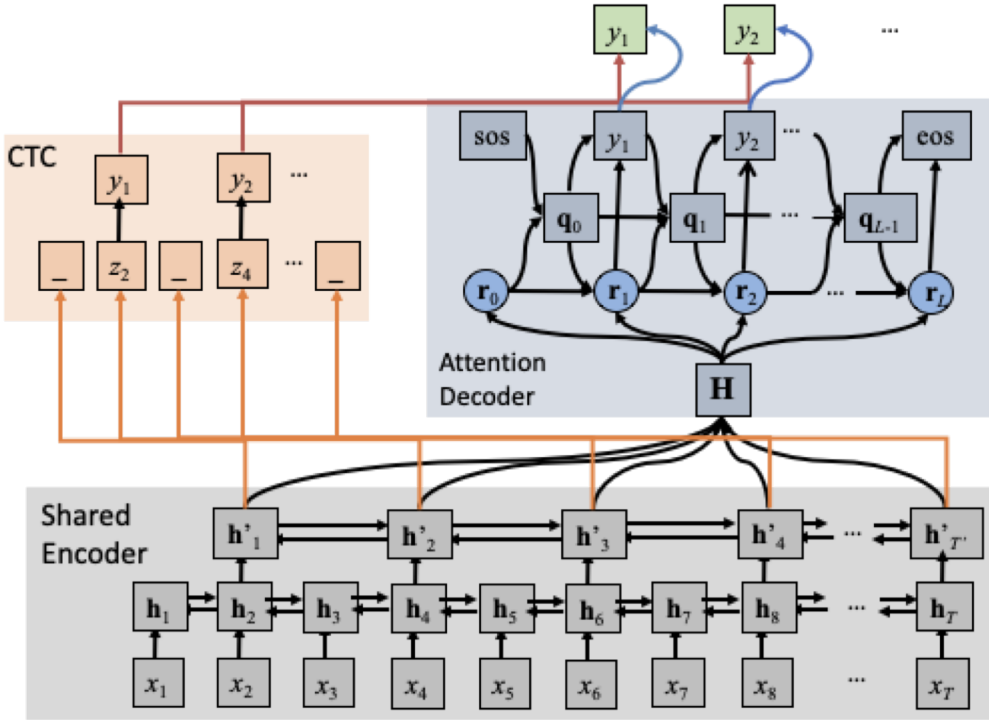


Joint CTC/attention (Joint C/A) [Kim+ 2017, Hori+ 2017]

- Combine CTC and attention during
  - training based on multi-task learning
  - inference based on score combination

😄 Very good performance with reasonable alignment

😓 Complicated implementation, off-line, slow, limited applications

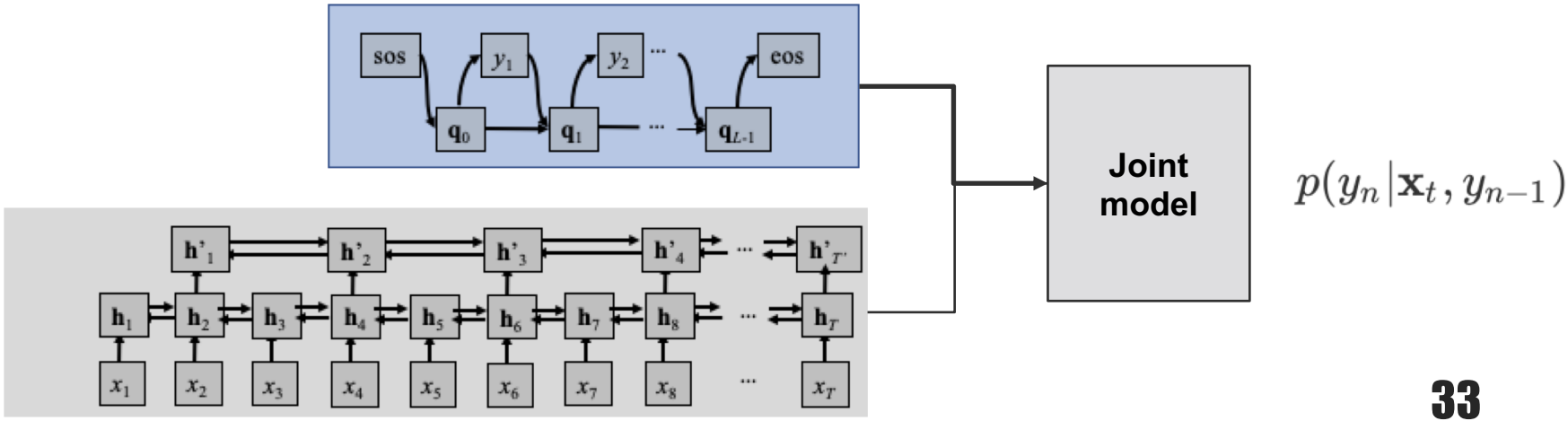


RNN-transducer [Graves+ 2013]

- Extension of CTC by considering previous output dependency
- Combine input RNN and auto-regressive output RNN to provide a joint distribution
  - Joint model can handle this combination

😄 Good performance with reasonable alignment, on-line

😓 Complicated implementation, slow, limited applications

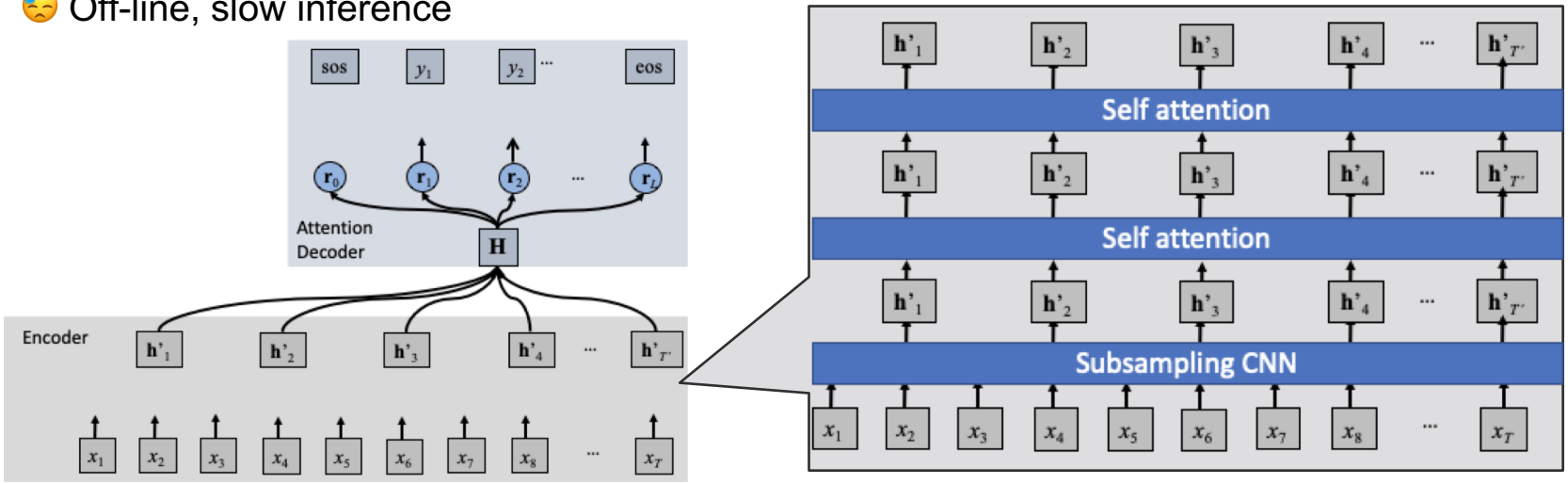


Transformer [Vaswani+ 2017, Dong+ 2018]

- Replace all recurrent connections in attention-based encoder-decoder with a self attention block (can capture very long-range dependency)
- All operations across time is well parallelized

😄 Very good performance with reasonable alignment, fast training, a lot of applications (ASR, TTS, NMT), relatively simple implementation

😓 Off-line, slow inference



Comparisons (based on my personal experience)

- **Performance**:

  Transformer > Joint C/A ~ RNN-T > ATT > CTC

- **Training and inference speed**:

  CTC > ATT > Joint C/A ~ RNN-T > Transformer
  (transformer's training is fast)

- **Online:**

  CTC, RNN-T > Joint C/A, ATT, Transformer

- **Application:**

  ATT~Transformer~RNN-T > CTC, Joint C/A

- **Easiness of implementation:**

  CTC > ATT > Transformer > Joint C/A > RNN-T

Unified view

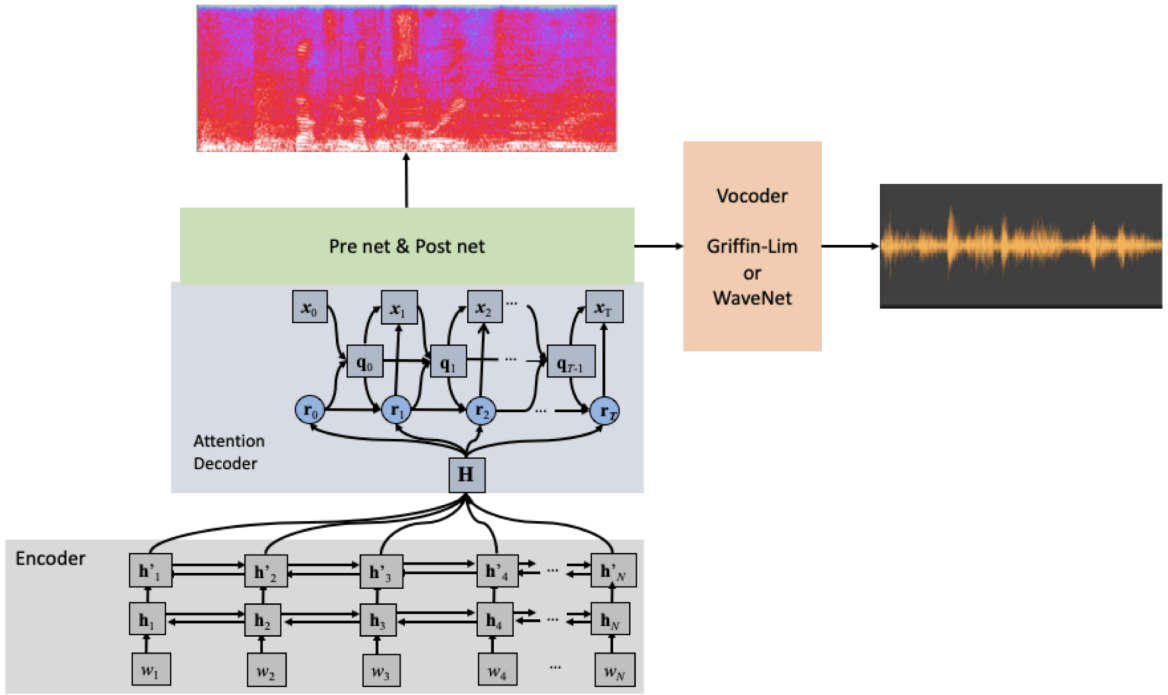
- Target speech processing problems: find a mapping function from *sequence* to *sequence* (**unification**)

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

  - ASR: X = Speech, Y = Text
  - TTS: X = Text, Y = Speech
  - ...
- Mapping function (f)
  - Attention based encoder decoder
  - Transformer
  - ...

Seq2seq TTS (e.g., Tacotron2) [Shen+ 2018]

- Use seq2seq generate a spectrogram feature sequence
- We can use either attention-based encoder decoder or transformer



Unified view → Unified software design

We design a new speech processing toolkit based on

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

Unified view → Unified software design

We design a new speech processing toolkit based on

$X = (x_1, x_2, \cdots, x_T)$

$Y = (y_1, y_2, \cdots, y_N)$

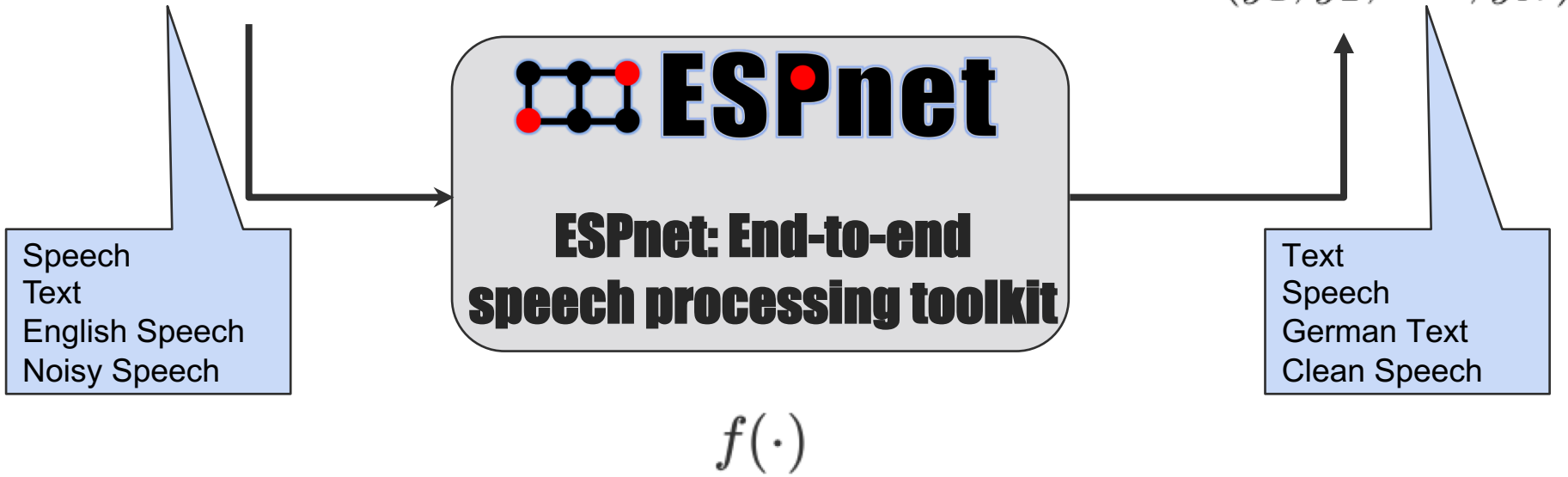
$f(\cdot)$

Unified view → Unified software design

We design a new speech processing toolkit based on

$X = (x_1, x_2, \cdots, x_T)$

$Y = (y_1, y_2, \cdots, y_N)$



Unified view → Unified software design

We design a new speech processing toolkit based on

$X = (x_1, x_2, \cdots, x_T)$

$Y = (y_1, y_2, \cdots, y_N)$

Unified view → Unified software design

We design a new speech processing toolkit based on

$X = (x_1, x_2, \cdots, x_T)$

$Y = (y_1, y_2, \cdots, y_N)$

- Many speech processing applications can be **unified** based on seq2seq
  - **Section 3** describes the **unified implementation** of several speech applications based on ESPnet

Table of contents

Seq2seq problem

How to deal with this sequence to sequence problem in a conventional approach?

$$X = (x_1, x_2, \cdots, x_T) \xrightarrow{f} Y = (y_1, y_2, \cdots, y_N)$$

Classical method (1970-)

$$\arg\max_{W} p(W|X)$$

X: Speech sequence
W: Text sequence

Classical method (1970-)

$$\arg\max_{W} p(W|X) = \arg\max_{W} p(X|W)p(W)$$
$$\approx \arg\max_{W,L} p(X|L)p(L|W)p(W)$$

L: Phoneme sequence

- **Speech recognition**
  - $p(X|L)$: Acoustic model (Hidden Markov model)
  - $p(L|W)$: Lexicon
  - $p(W)$: Language model (n-gram)

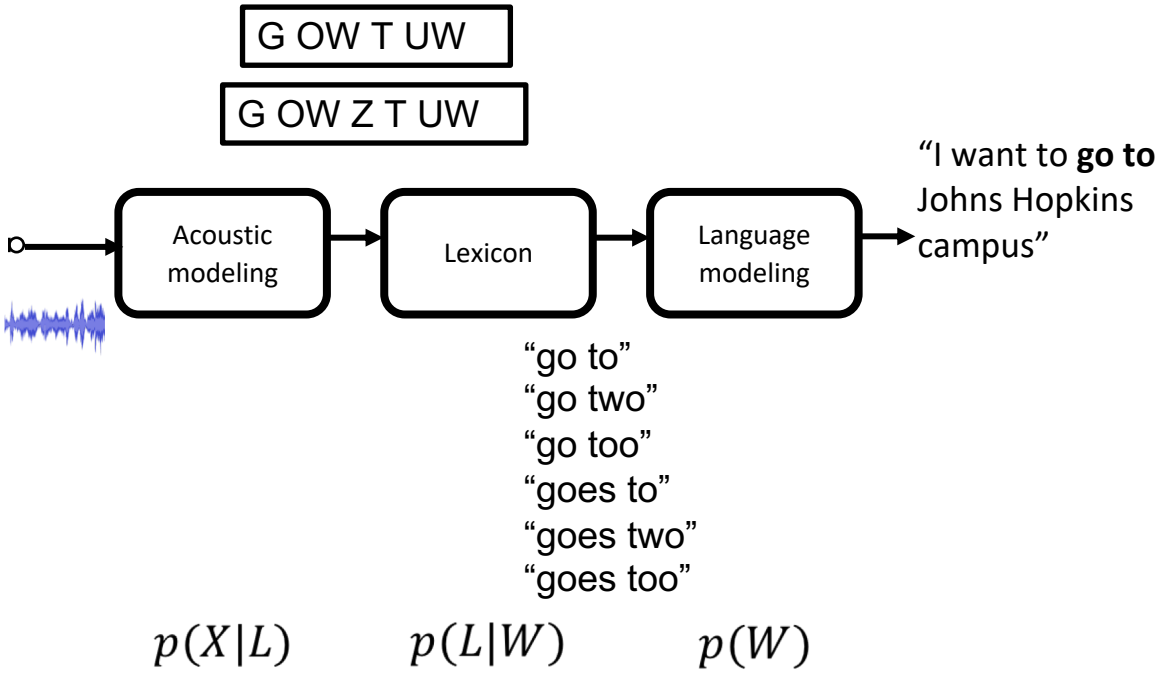
Classical method (1970-)

$$\arg\max_{W} p(W|X) = \arg\max_{W} p(X|W)p(W)$$
$$\approx \arg\max_{W,L} p(X|L)p(L|W)p(W)$$

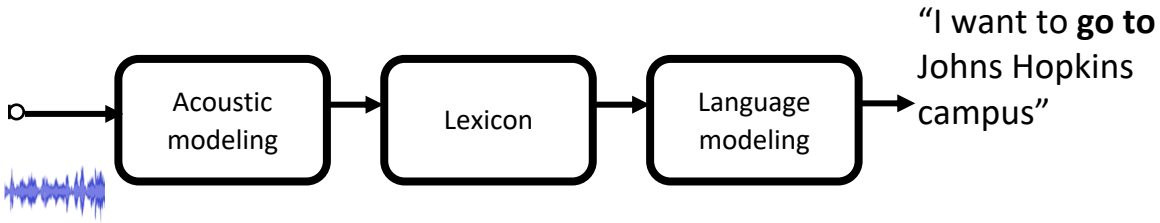
- **Speech recognition**
  - $p(X|L)$: Acoustic model (Hidden Markov model)
  - $p(L|W)$: Lexicon
  - $p(W)$: Language model (n-gram)

Modularization
based on probabilistic factorization

Speech recognition pipeline

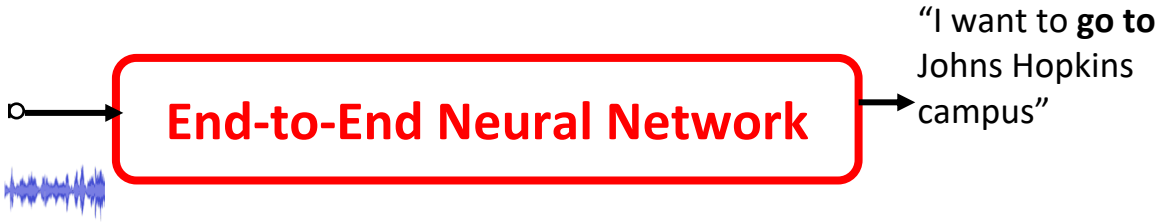


Speech recognition pipeline



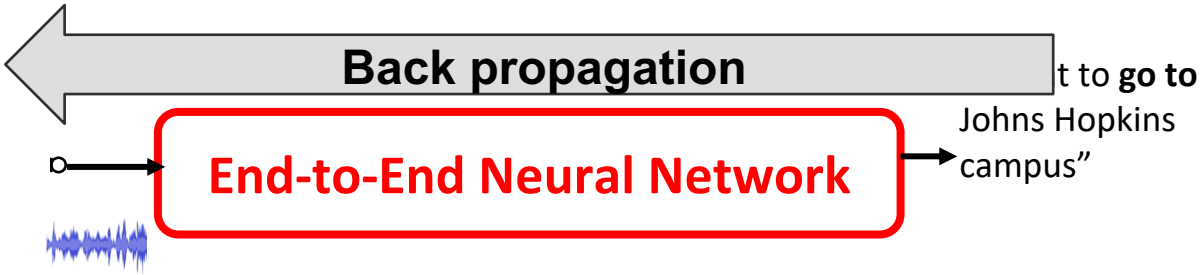
- Modular-based integration
- Require **a lot of development** for an acoustic model, a pronunciation lexicon, a language model, and finite-state-transducer decoding
- Require linguistic resources
- Difficult to build ASR systems for non-experts

From pipeline to integrated architecture



- Greatly simplify the complicated model-building/decoding process
- Easy to build ASR systems for new tasks **without expert knowledge**
- Train a deep network that directly maps speech signal to the target letter/word sequence
- **Integrate acoustic, lexicon, and language models**

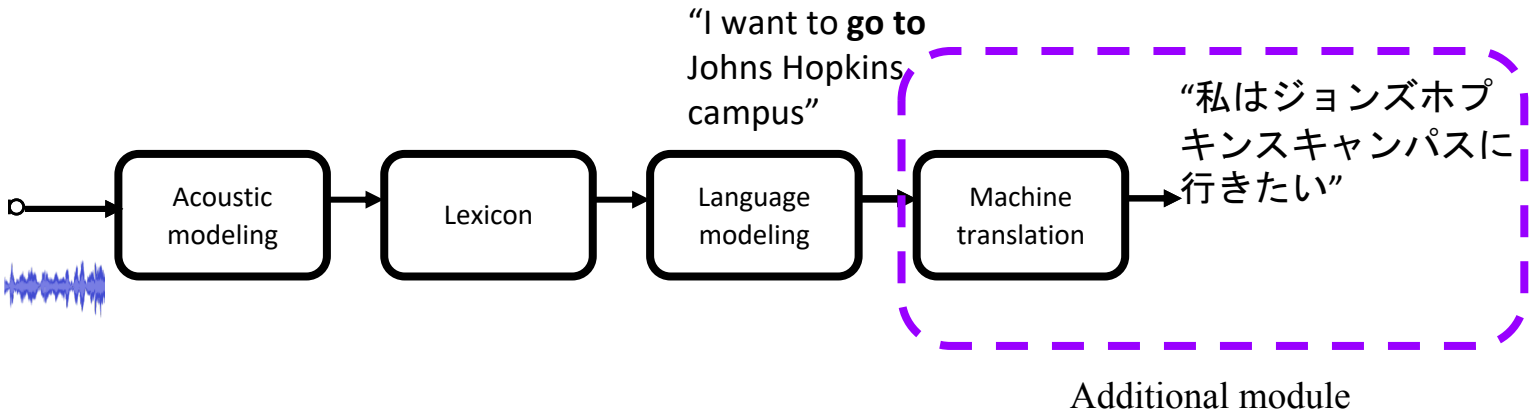
From pipeline to integrated architecture



- Greatly simplify the complicated model-building/decoding process
- Easy to build ASR systems for new tasks **without expert knowledge**
- Train a deep network that directly maps speech signal to the target letter/word sequence
- **Integrate acoustic, lexicon, and language models**
- Potentially optimize entire ASR network by **back propagation**

Integration

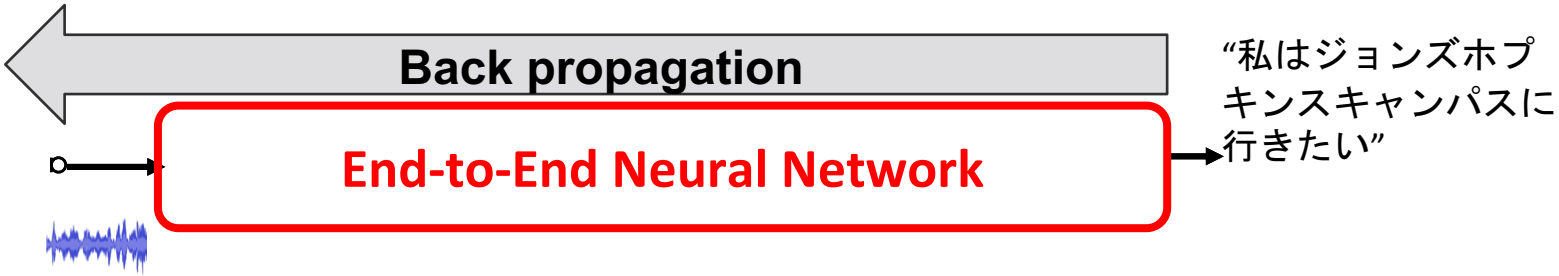
Speech translation example



We can further integrate other speech processing modules

Integration

We can further integrate other speech processing modules



- End-to-end neural architecture allows us to **integrate** multiple speech processing applications
  - **Section 2** describes several **integration** examples

Summary of this section

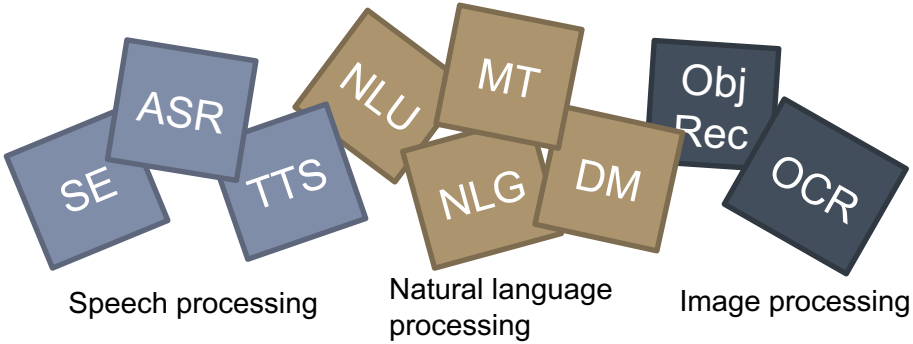
- Many speech processing applications can be **unified** based on seq2seq
  - **Section 3** describes the **unified implementation** of several speech applications based on ESPnet
- End-to-end neural architecture allows us to **integrate** multiple speech processing applications
  - **Section 2** describes several **integration** examples

2. End-to-End Integration of Multiple Speech Applications

Takaaki Hori
Mitsubishi Electric
Research Laboratories
(MERL)

2.1 Introduction

- Various speech technologies are available as software libraries, which can be combined with other technologies

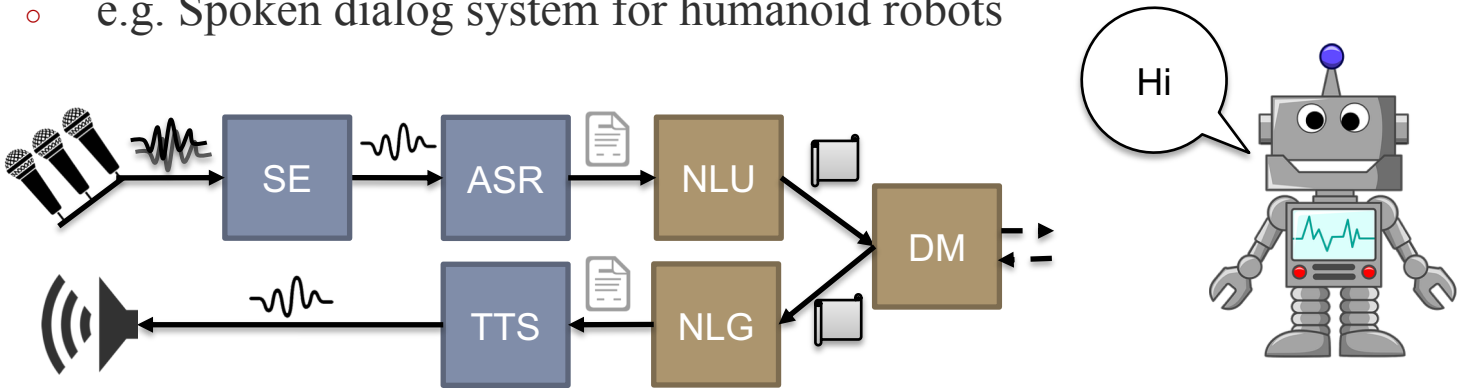


- What can we do by combining these technologies?
  - (1) Build advanced speech applications beyond pure speech processing
  - (2) Boost speech technology using other technologies

This section reviews recent studies on neural end-to-end integration

2.2 (Integration 1) Build advanced speech applications

- Purpose of integration is typically to build a new application or add a new functionality to an existing application
  - e.g. Spoken dialog system for humanoid robots



- Software development kit (SDK) helps application engineers equip various speech technologies to the target application
- Complex speech applications can be created more easily

Example applications	SE	ASR	TTS	OCR	MT	NLU/DM
Digital hearing aid  Audicus Clara	H					
Dictation system  IBM ViaVoice (1997~)		H				
Text-to-speech system  DecTalk (1984)			H			
Text-to-speech reader  C-Pen Reader			H	H		
Speech translator  Pocketalk (2018)		H	H		H	
Car navigation system  Mitsubishi (NR-MZ300PREMI)	H	H	H			H
Smartphone  iPhone + Siri (2011)		H	H			H
Smart speaker   Amazon Echo (2014) / Google Home (2016)	H	H	H			H
Humanoid robot  Honda ASIMO (2000~)  Softbank Pepper (2014)	H	H	H			H

Modular-based integration

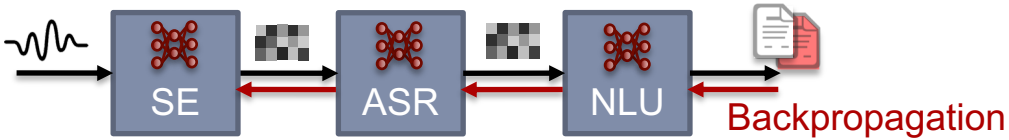
- Build and train each module independently
- Combine multiple modules through dedicated APIs

(e.g. Noise-robust speech understanding)

- Pros
  - Easy to design and build a system
  - Modules are reusable for other applications
- Cons
  - Difficult to resolve mismatches between (blackbox) modules
  - Need intensive tuning of each module
  - Errors can be escalated through the pipeline

Neural end-to-end integration

- Build each module as a neural network
- Combine the multiple neural networks and train them jointly

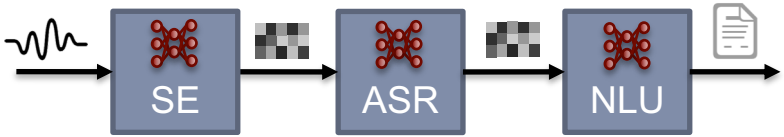


(e.g. Noise-robust speech understanding)

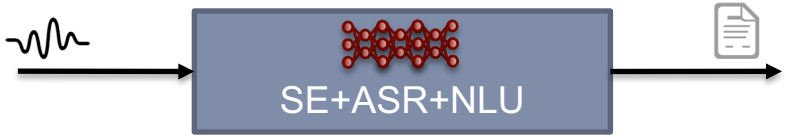
- Pros
  - Can optimize the entire system to minimize the output error
  - Can reduce mismatches between networks by joint optimization
  - Does not require intermediate target signals/labels for training
- Cons
  - Networks are interdependent. Cannot be used for other applications

Cascade vs. monolithic architecture

- Cascade architecture
  - Explicitly define a role of each network and combine them
  - Existing architecture can be borrowed to design a system

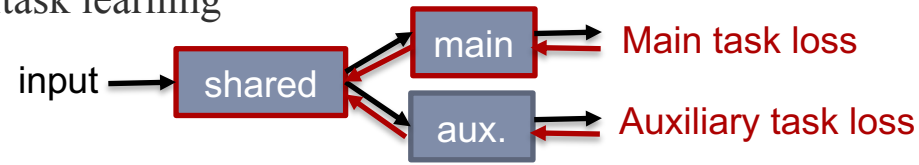


- Monolithic architecture
  - Design an entire network without considering the explicit role of each network component
  - Potential to make the model simple and effective for the target application by tighter coupling of multiple networks

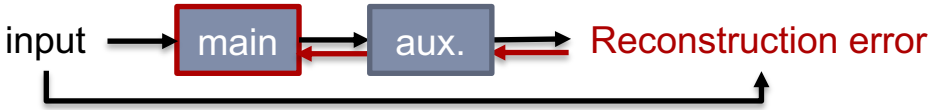


2.3 [Integration-2] Boost speech technology

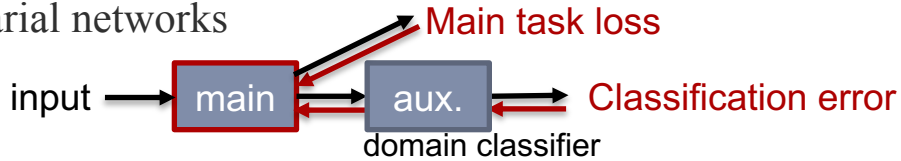
- Neural end-to-end integration can be used to boost a main network with an auxiliary network
  - Using an auxiliary network for a similar task
    - Multitask learning



  - Using an auxiliary network with an inverse process (e.g. ASR⇄TTS)
    - Cycle-consistency unsupervised training



  - Using an auxiliary network for an adversarial task
    - Adversarial networks



Improved training with auxiliary networks

- Multi-task learning
  - Speech translation network is trained with ASR network [Weiss+ 2017]
  - End-to-end SLU network is trained with ASR network [Haghani+ 2018]

- Cycle consistency un-/semi-supervised training
  - TTS is used to train ASR network [Tjandra+ 2017, Hayashi+ 2018, Hori+ 2019, Baskar+ 2019]
  - ASR is used to train TTS network [Tjandra+ 2017, Liu+ 2018]

- Adversarial training
  - Speaker recognition is used to make an acoustic model more robust against speaker variations [Saon+ 2017, Meng+ 2018]

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR) ESPnet
b. Multispeaker speech recognition (SS + ASR) ESPnet
c. Multichannel speech recognition (SE + ASR) ESPnet
d. Speech translation (ASR + MT (+TTS)) ESPnet
e. Spoken language understanding (ASR + NLU)

(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models
   - Cycle-consistency training (ASR + TTS) ESPnet
g. Other semi-supervised approaches
h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

ESPnet
ESPnet supports its recipe

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. **Multilingual speech recognition (LID + ASR)** ESPnet
b. Multispeaker speech recognition (SS + ASR) ESPnet
c. Multichannel speech recognition (SE + ASR) ESPnet
d. Speech translation (ASR + MT (+TTS)) ESPnet
e. Spoken language understanding (ASR + NLU)

(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models ESPnet
   - Cycle-consistency training (ASR + TTS)

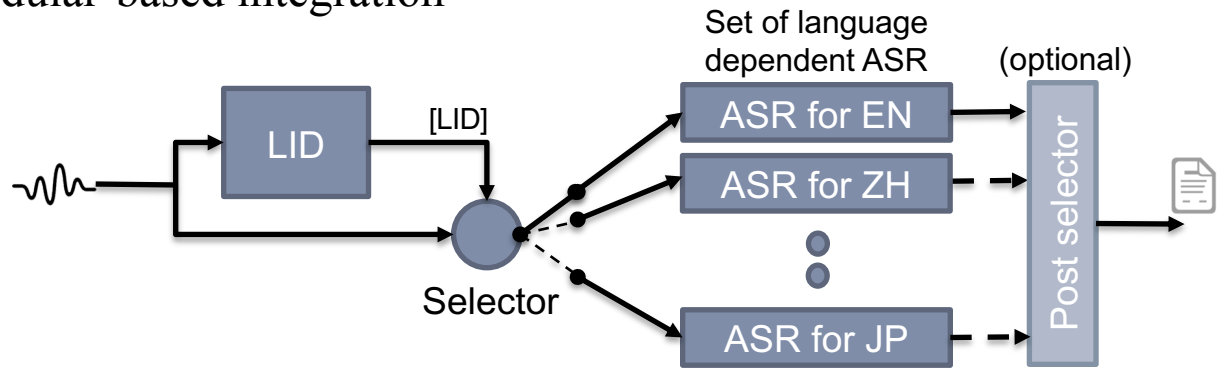
g. Other semi-supervised approaches
h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

ESPnet
ESPnet supports its recipe

a. Multilingual speech recognition

Language identification (LID) + Multilingual ASR

- Modular-based integration

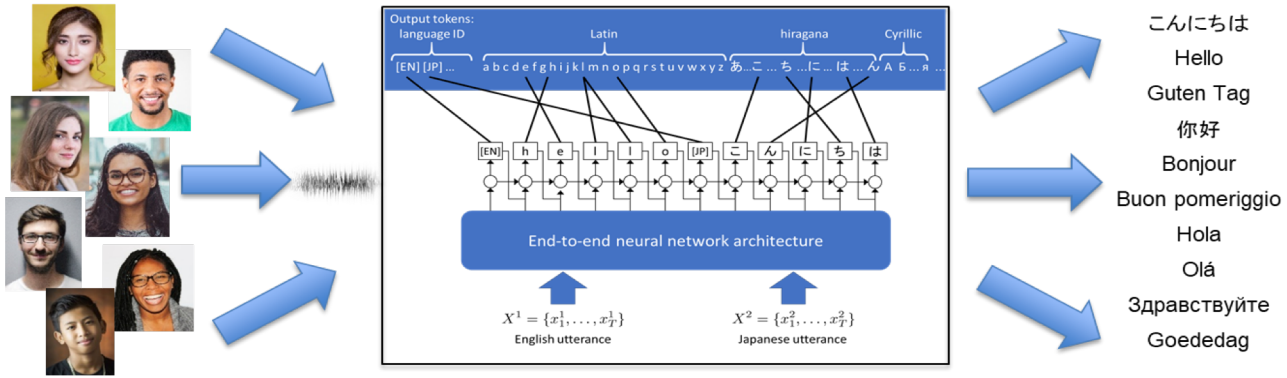


(ASR systems may share acoustic models and search space [Gonzalez-Dominguez+ 2015])

- Neural end-to-end integration
  - Monolithic architecture for LID and ASR

Multilingual end-to-end ASR

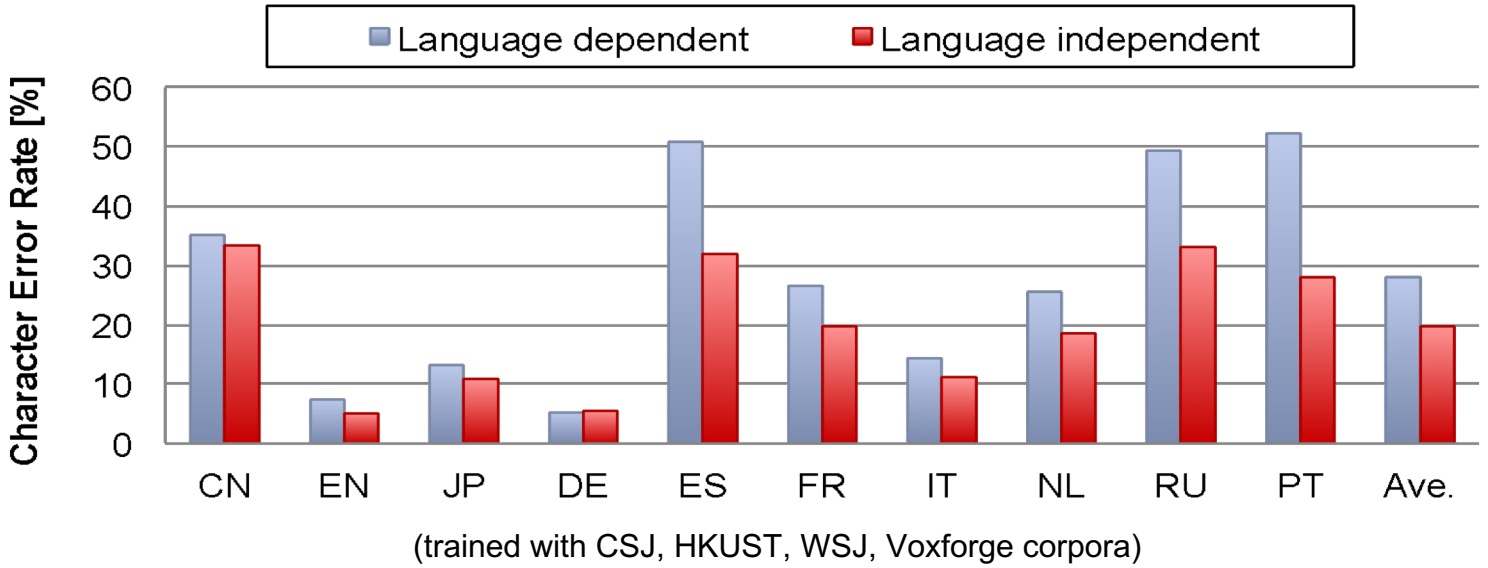
- End-to-end network for joint language identification and ASR [Watanabe+ 2017, Kim+ 2018, Toshniwal+ 2018]
  - Learn a single network using multilingual speech corpora, appending a language ID label to each utterance transcript
    ("*[EN] as signs of a stronger economy emerge he adds long term …*")



- Learn the network with code-switching data [Seki+ 2018]
  ("*[EN] hello [JP] こんにちは [ZH] 您好 …*")

ASR performance for 10 languages [Watanabe+ 2017b]

- Comparison with language dependent systems



(trained with CSJ, HKUST, WSJ, Voxforge corpora)

- Reduced errors in most languages by the unified network
- Framework has been tested up to 100 languages [Oliver+ 2019]

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR)

b. Multispeaker speech recognition (SS + ASR)

c. Multichannel speech recognition (SE + ASR)

d. Speech translation (ASR + MT (+TTS))

e. Spoken language understanding (ASR + NLU)

(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models
   - Cycle-consistency training (ASR + TTS)

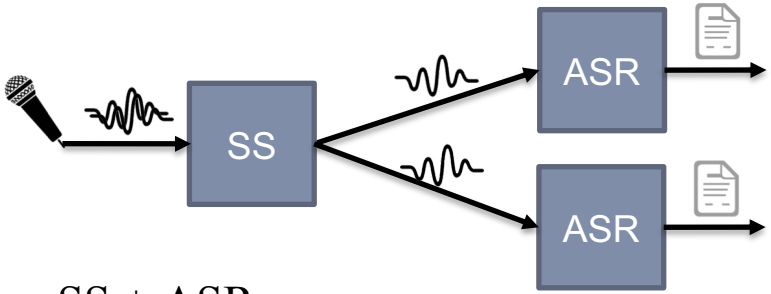
g. Other semi-supervised approaches

h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

ESPnet supports its recipe

b. Multispeaker speech recognition

Single-channel speech separation (SS) is useful to transcribe overlapped utterances in single-channel recordings



Prior work on SS + ASR

- Deep clustering + ASR (retraining helps reduce the mismatch) [Isik+ 2016]
- Permutation invariant acoustic model (AM) training, where AM has both SS and phone recognition capabilities [Qian+ 2017]

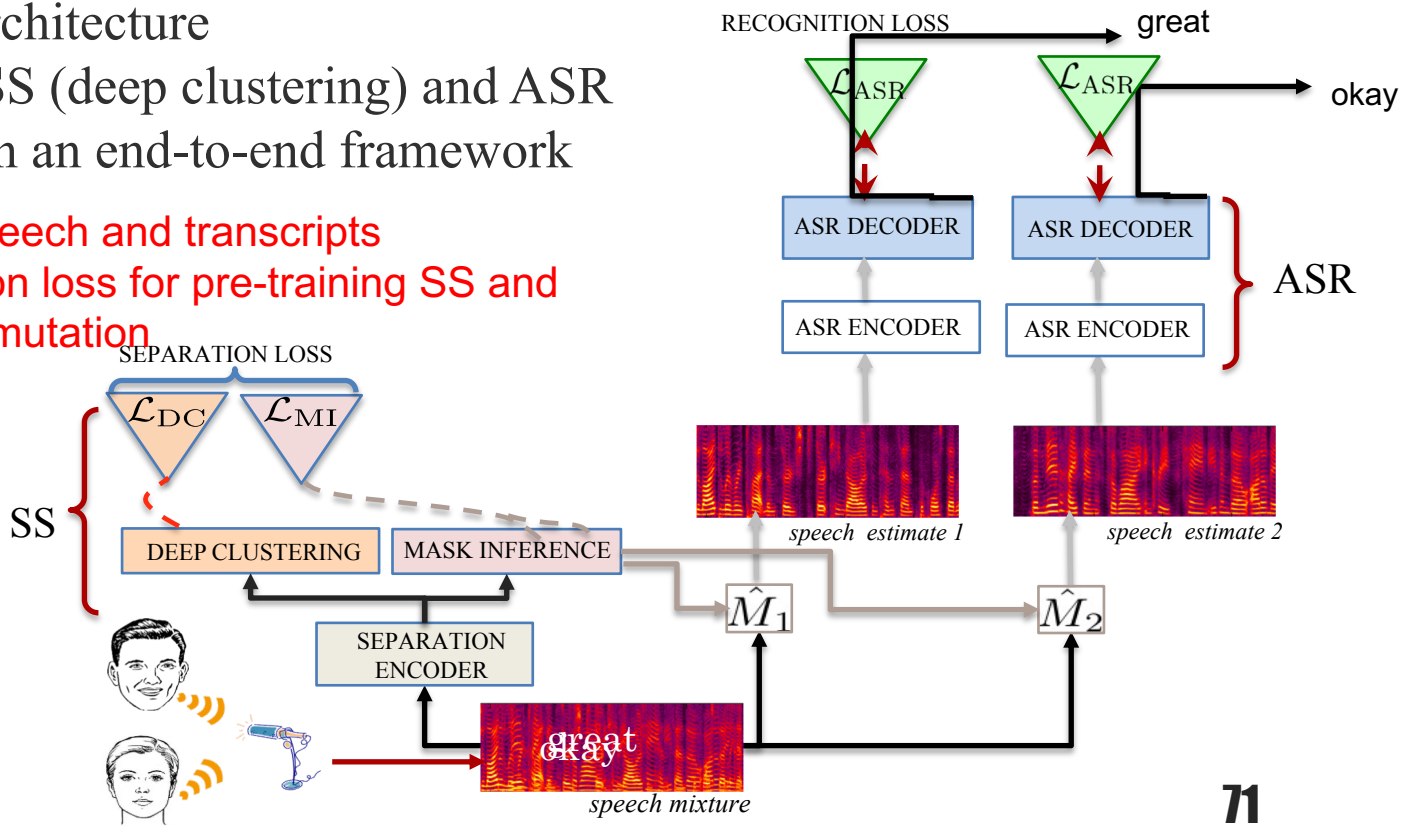
End-to-end approaches

- Cascade architecture for SS+ASR [Shane+ 2018]
- Monolithic architecture for SS+ASR [Seki+ 2018]

End-to-end speech separation & recognition [Shane+ 2018]

- Cascade architecture
- Combine SS (deep clustering) and ASR networks in an end-to-end framework

- Use target speech and transcripts
- Use separation loss for pre-training SS and resolving permutation



Joint separation & recognition experiments [Shane+ 2018]

Oracle and baseline CER results (%) (w/ char LM)

Training	Test	eval CER (%)
CLN	CLN	6.6
IBM	IBM	9.0
CLN	MIX	79.1

Proposed method, CER results (%) (w/ char LM)

Fine-tuning			CLN-ASR-PT		IBM-ASR-PT	
SS	ASR	Loss	dev CER (%)	eval CER (%)	dev CER (%)	eval CER (%)
NO	NO	-	34.1	32.0	24.2	23.1
NO	YES	ASR	18.9	18.0	18.7	17.9
YES	YES	SS+ASR	16.3	15.4	14.0	13.9
YES	YES	ASR	**13.3**	**13.2**	13.6	13.4

Purely end-to-end approach [Seki+ 2018, Chang+ 2019]

ESPnet

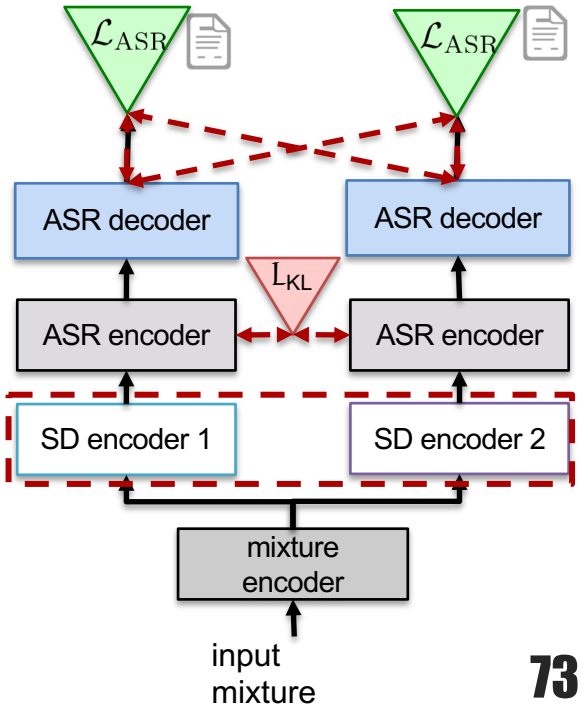
- Monolithic architecture without any explicit separation network
- Incorporate implicit separation via speaker-differentiating (SD) encoders followed by a shared recognition encoder
- Transcript-level permutation-free loss

$$\mathcal{L} = \min_{\pi \in \mathcal{P}} \sum_{s=1}^{S} \mathcal{L}_{\mathrm{ASR}}(Y^S, R^{\pi(s)})$$

S: number of speakers	Y: network output
$\mathcal{P}$: possible permutations	R: reference

- Additional KL loss helps implicit separation
- No need for target speech in training

Resolve permutation and backprop



Purely end-to-end approach [Seki+ 2018, Chang+2019]

ESPnet

- CER (%) of mixed speech for WSJ (w/ word LM)

	High E. spk.	Low E. spk.	Avg.
Baseline	86.4	79.5	83.0
Purely E2E model	14.6	13.3	14.0
+KL loss	**14.0**	**13.3**	**13.7**

- Comparison with other methods

	CER (%)	WER (%)
Deep Clustering + ASR pipeline [Isik+'16]	-	30.8
Cascade E2E model [Settle+'18]	13.2	-
Purely E2E model (+KL loss)	**14.0**	**28.2**

  - Competitive performance with cascaded SS+ASR network
  - No need for target speech in training

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR)
b. Multispeaker speech recognition (SS + ASR)
c. **Multichannel speech recognition (SE + ASR)**
d. Speech translation (ASR + MT (+TTS))
e. Spoken language understanding (ASR + NLU)

(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models
   - Cycle-consistency training (ASR + TTS)
g. Other semi-supervised approaches
h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

ESPnet supports its recipe

c. Multichannel speech recognition

Beamformer (BF) with a microphone array is an important frontend to enhance the target speech in distant-talk ASR scenario.

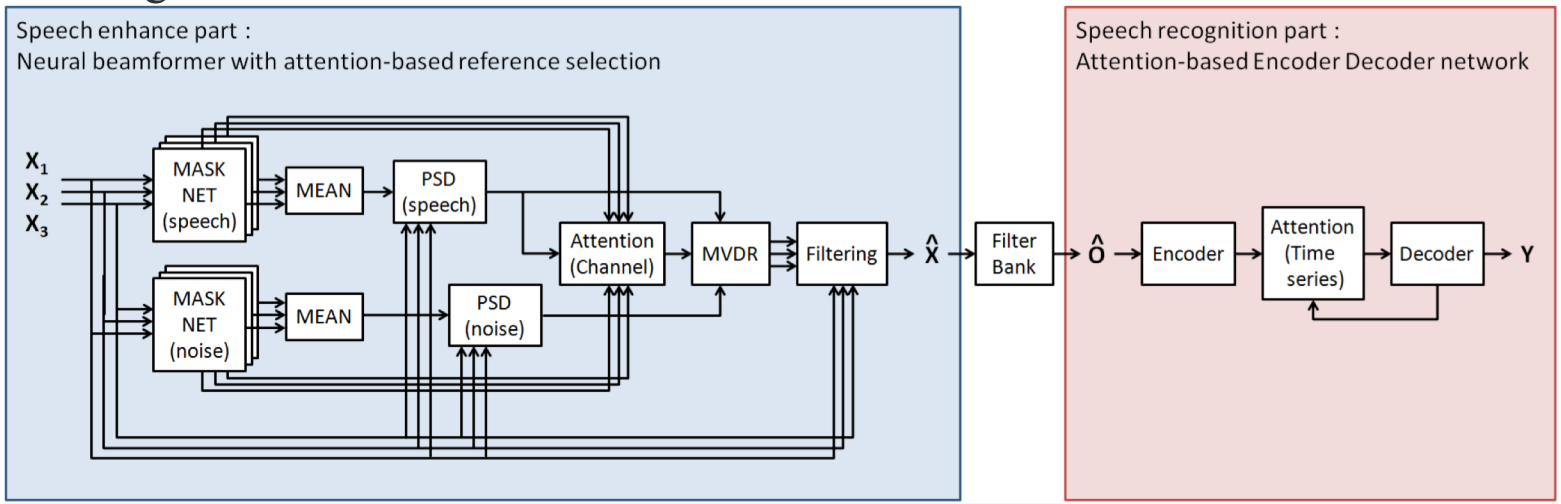
Performance degradation due to the mismatch between the enhanced speech and the acoustic model trained with clean speech

Approaches to reduce the mismatch:

- Joint beamformer and acoustic model training [Sainath+ 2016]
  ⇨ CNN (as a filter-and-sum beamformer) + CLDNN acoustic model
- Neural beamformer + acoustic model training [Xiao+ 2016]
- Neural beamformer + end-to-end ASR [Ochiai+ 2017]

End-to-end beamforming and ASR [Ochiai+ 2017]

- BF: Neural beamformer with attention-based reference selection
- ASR: Attention-based encoder-decoder network
- Cascade architecture of BF and ASR outperforms conventional pipeline integration



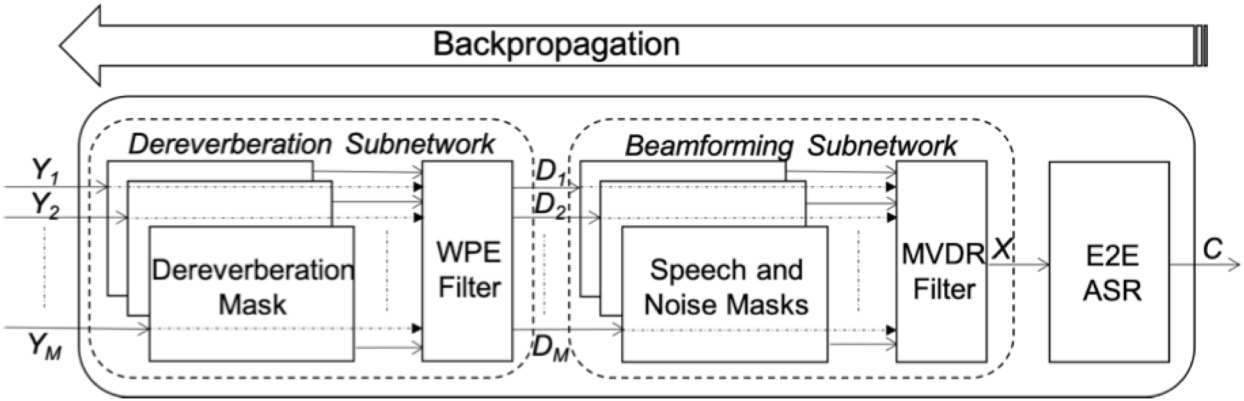
Ochiai, Tsubasa, et al. "Unified architecture for multichannel end-to-end speech recognition with neural beamforming." IEEE Journal of Selected Topics in Signal Processing 11.8 (2017): 1274-1288.

End-to-end dereverberation, beamforming, and ASR
[Subramanian+ 2019]

Cascade architecture including Derev., BF, and ASR

ESPnet

- Derev.: DNN-WPE, BF: MaskNet-MVDR, ASR: CTC-attention



Subramanian, Aswin Shanmugam et al., "An investigation of end-to-end multichannel speech recognition for reverberant and mismatch conditions, arXiv preprint, 1904.09049v3, 2019

- Joint training makes the model more robust to mismatched conditions, and provides comparable or better performance than existing pipeline methods

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR) ESPnet
b. Multispeaker speech recognition (SS + ASR) ESPnet
c. Multichannel speech recognition (SE + ASR) ESPnet
d. **Speech translation (ASR + MT (+TTS))** ESPnet
e. Spoken language understanding (ASR + NLU)

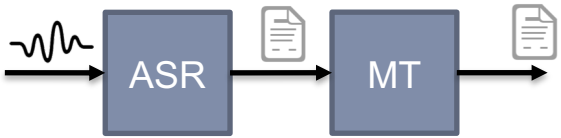
(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models ESPnet
   - Cycle-consistency training (ASR + TTS)
g. Other semi-supervised approaches
h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

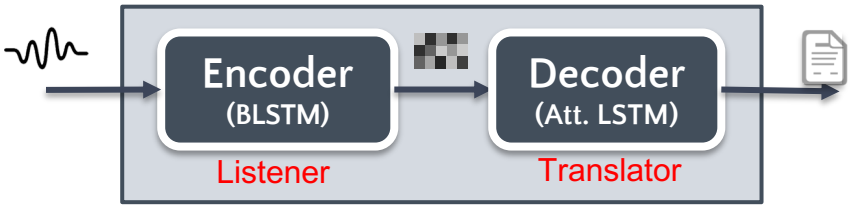
ESPnet
ESPnet supports its recipe

d. Speech translation

- Combining ASR and MT enables speech-to-text translation
- Both ASR and MT models can be built as a neural network
- Neural end-to-end speech translation is possible



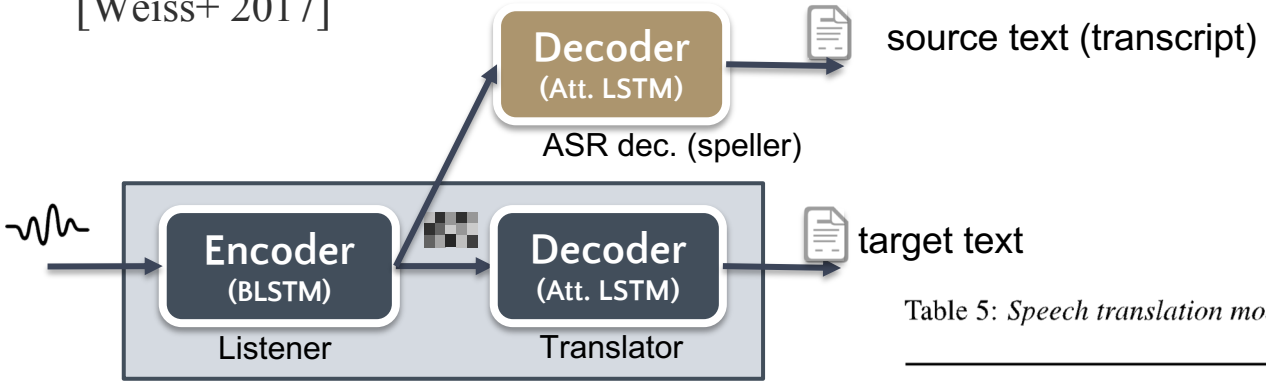
- "Listen and translate" approach [Berard+ 2016]
  - Use a monolithic encoder decoder for speech translation ESPnet



No need for source text

Multi-task learning for speech translation

- ASR decoder is used as an auxiliary network connected with the shared encoder [Weiss+ 2017]



- Encoder more focuses on ASR encoding
- Decoder more focuses on MT
- Need source text but better performance

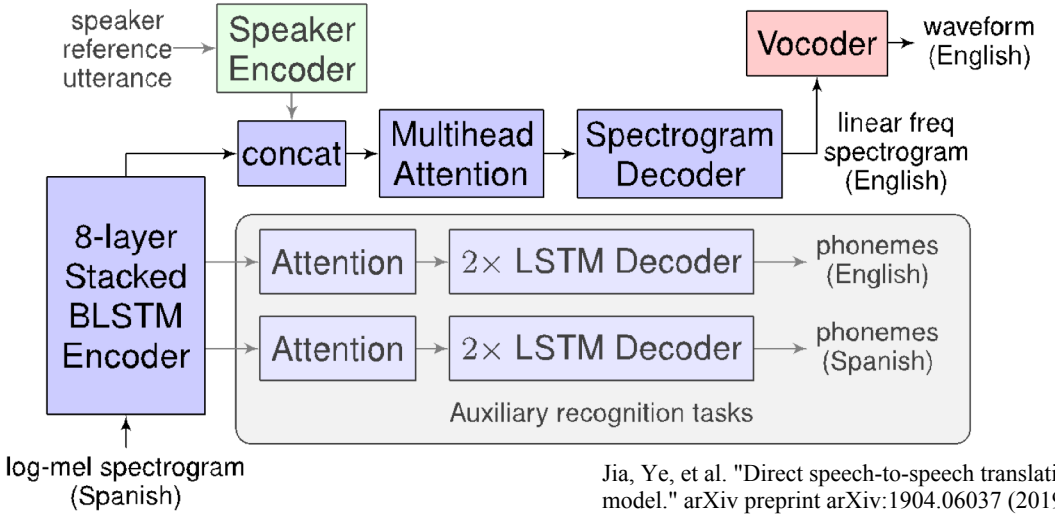
Table 5: *Speech translation model performance in BLEU score.*

Model	Fisher dev	Fisher dev2	Fisher test	Callhome devtest	Callhome evltest
End-to-end ST[3]	46.5	47.3	47.3	16.4	16.6
Multi-task ST / ASR[3]	48.3	49.1	48.7	16.8	17.4
ASR→NMT cascade[3]	45.1	46.1	45.5	16.2	16.6
Post et al. [19]	–	35.4	–	–	11.7
Kumar et al. [21]	–	40.1	40.4	–	–

Weiss, Ron J., et al. "Sequence-to-Sequence Models Can Directly Translate Foreign Speech." Proc. Interspeech 2017 (2017): 2625-2629.

Speech-to-speech translation

- Direct speech-to-speech translation (S2ST) [Jia+2019]



Jia, Ye, et al. "Direct speech-to-speech translation with a sequence-to-sequence model." arXiv preprint arXiv:1904.06037 (2019).

- End-to-end S2ST model as a single attention-based encoder decoder
- Multi-task learning with auxiliary decoders for ASR and ST
- Slightly underperforms the cascade of ST and TTS, but it works!

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR) ESPnet
b. Multispeaker speech recognition (SS + ASR) ESPnet
c. Multichannel speech recognition (SE + ASR) ESPnet
d. Speech translation (ASR + MT (+TTS)) ESPnet
e. **Spoken language understanding (ASR + NLU)**

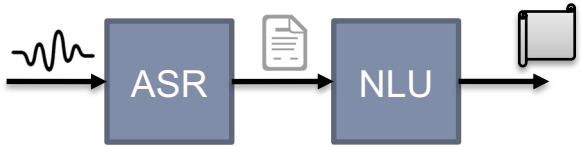
(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models ESPnet
   - Cycle-consistency training (ASR + TTS)
g. Other semi-supervised approaches
h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

ESPnet
ESPnet supports its recipe

e. Spoken language understanding

- Predict semantic information from speech



- NLU: Sentence text ⇨ Semantic representation accepted by the backend of the application (e.g., set of slot-value pair)

"set an alarm for 6 p.m." ⇨

```
Domain: PRODUCTIVITY
Intent: SET_ALARM
data_time: 6 p.m.
```

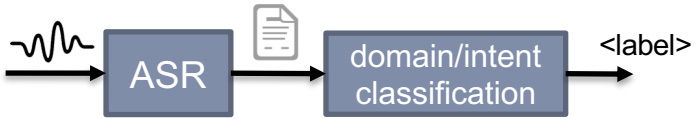
- Conventional NLU pipeline

End-to-end spoken language understanding

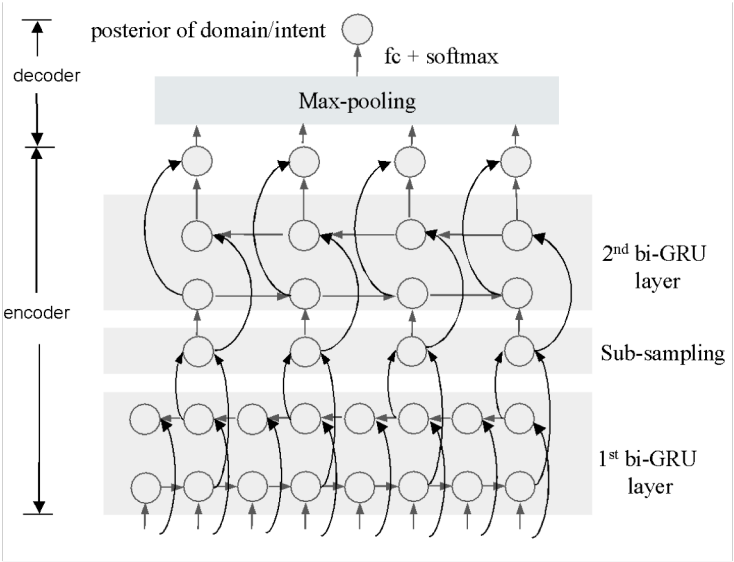
[Serdyuk+ 2018]

First work on end-to-end SLU

- Consider only domain / intent classification (speech-to-single-label)



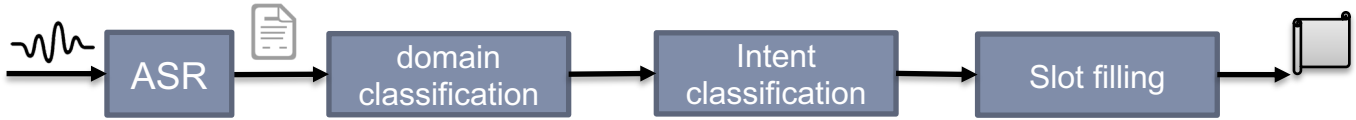
- Monolithic speech encoder-single label prediction network
  - No significant improvement from a conventional ASR-NLU pipeline
  - But no need for transcription during training, and
  - E2E model is more compact: Training and inference are faster due to no ASR decoding



Serdyuk, Dmitriy, et al. "Towards end-to-end spoken language understanding." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

Sequence-to-sequence model for end-to-end SLU
[Haghani+ 2018]

- Consider all functions for SLU



- Use an attention-based encoder decoder that directly converts speech to serialized semantic representation, i.e., sequence-to-sequence problem

Domain: PRODUCTIVITY
Intent: SET_ALARM
data_time: 6 p.m.

Serialization

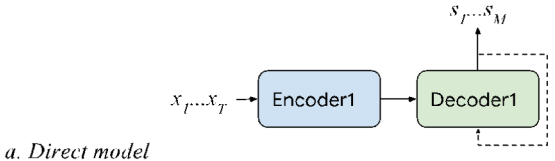
⇨ <DOMAIN><PRODUCTIVITY><INTENT><SET_ALARM><DATETIME>6 p.m.

Transcript	Serialized Semantics
"can you set an alarm for 2 p.m."	<DOMAIN><PRODUCTIVITY><INTENT><SET_ALARM><DATETIME>2 p.m.
"remind me to buy milk"	<DOMAIN><PRODUCTIVITY><INTENT><ADD_REMINDER ><SUBJECT>buy milk
"next song please"	<DOMAIN><MEDIA_CONTROL>
"how old is barack obama"	<DOMAIN><NONE>

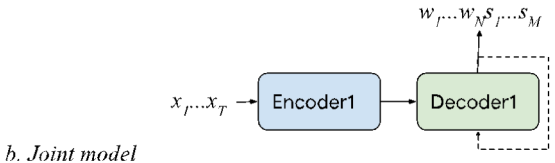
Haghani, Parisa, et al. "From Audio to Semantics: Approaches to end-to-end spoken language understanding." 2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018.

Sequence-to-sequence model for end-to-end SLU
[Haghani+ 2018]

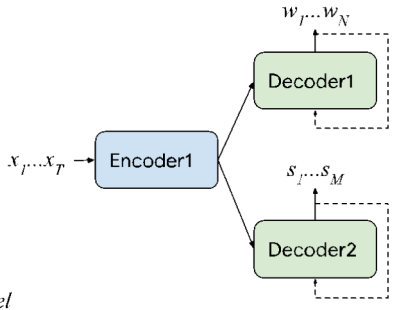
- End-to-end SLU models and their performance



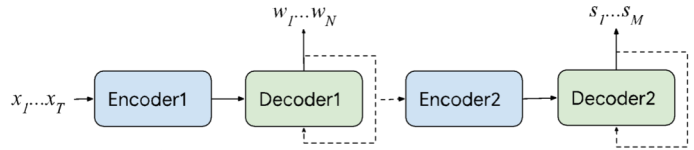
a. Direct model



b. Joint model



c. Multitask model



d. Multistage model

Table 4: *Domain and intent F1 scores, and argument WER for the predicted semantics.*

Model	Domain F1	Intent F1	Arg WER
Baseline	96.6	95.1	15.04
Direct	96.2	94.2	18.22
Joint	**96.8**	**95.7**	14.93
Multitask	**96.7**	**95.8**	15.02
Multistage (ArgMax)	96.5	95.4	14.84
Multistage (SampledSoftmax)	96.5	95.2	**12.29**

Haghani, Parisa, et al. "From Audio to Semantics: Approaches to end-to-end spoken language understanding." 2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018.

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR)
b. Multispeaker speech recognition (SS + ASR)
c. Multichannel speech recognition (SE + ASR)
d. Speech translation (ASR + MT (+TTS))
e. Spoken language understanding (ASR + NLU)

(Integration 2) **Boost speech technologies**

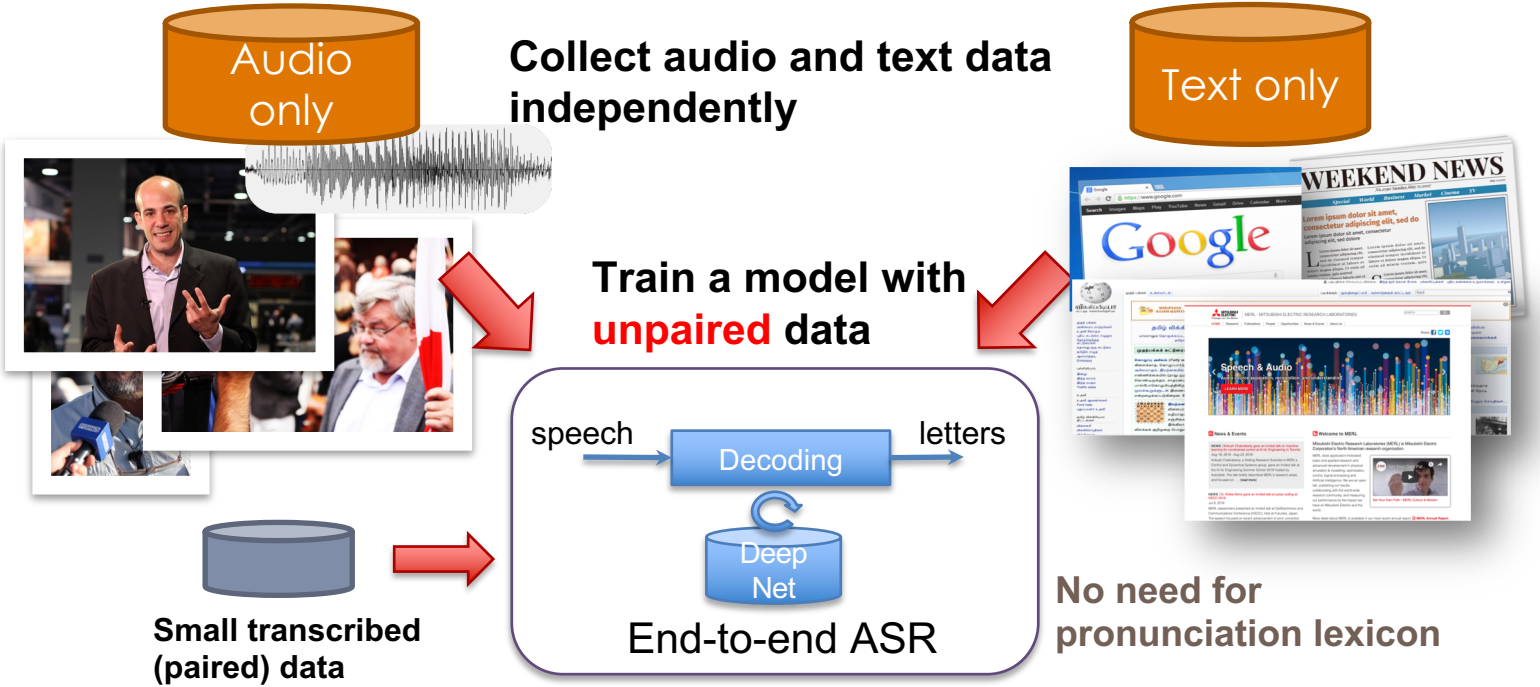
f. Semi-supervised training of ASR models
- **Cycle-consistency training (ASR + TTS)**

g. Other semi-supervised approaches
h. Domain adversarial training
- Speaker-invariant training (ASR + SR)

ESPnet supports its recipe

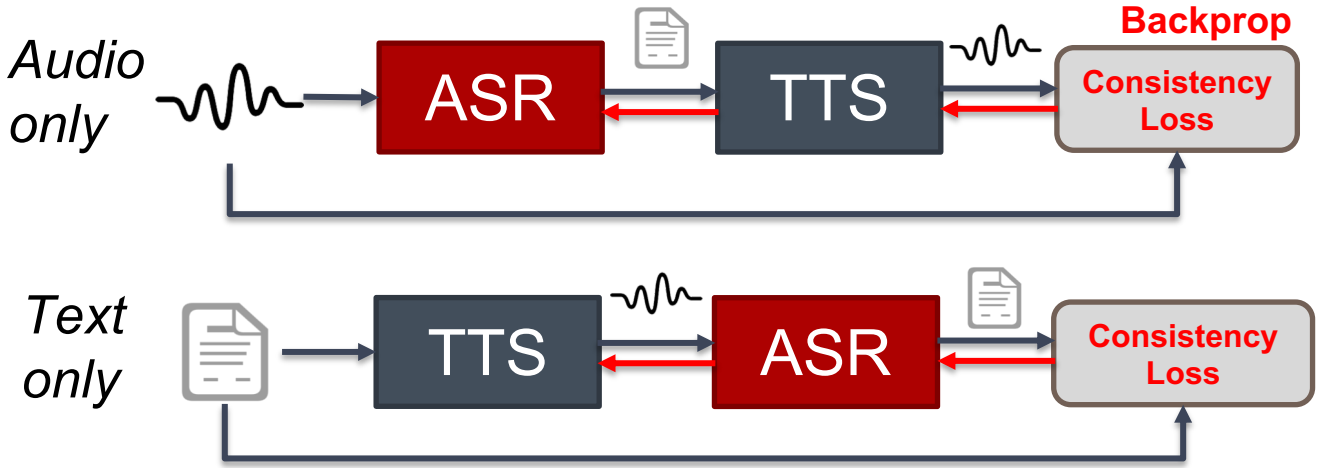
f. Semi-supervised training of ASR models

- Collecting paired data is very expensive to train ASR models
- Semi-supervised training with unpaired data reduces the development cost



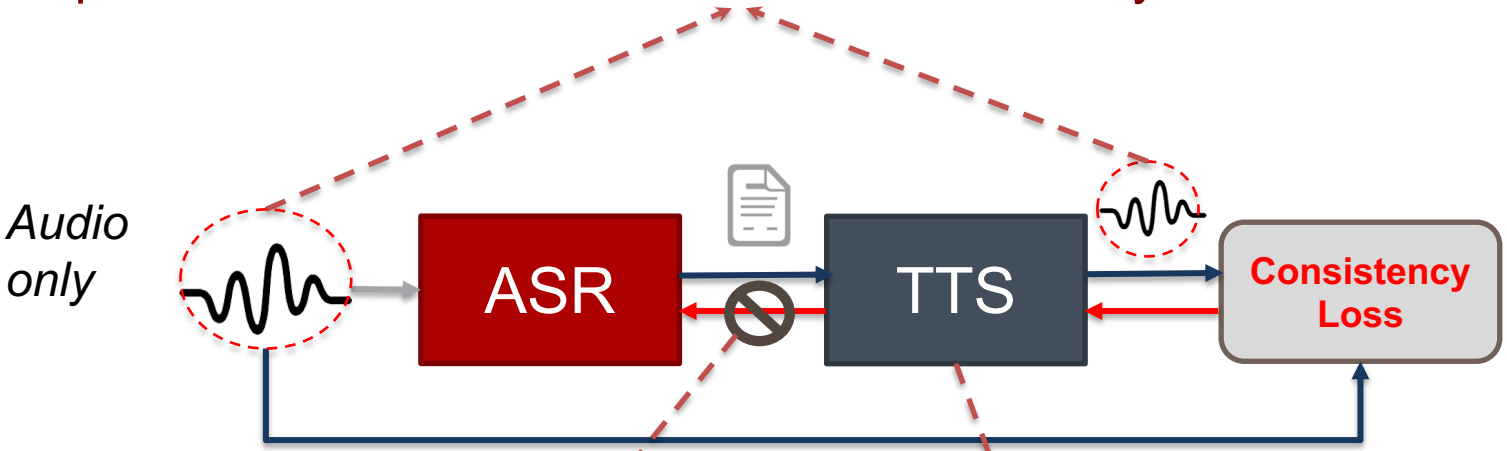
Cycle-consistency training with unpaired data

- **Cycle-consistency loss** (reconstruction error) was introduced in machine translation [He+ 2016] and image transformation [Zhu+ 2017].
- For ASR, neural end-to-end integration of ASR and TTS can be used



Problems in cycle-consistency training for ASR

1. Speaker mismatch deteriorates the consistency loss

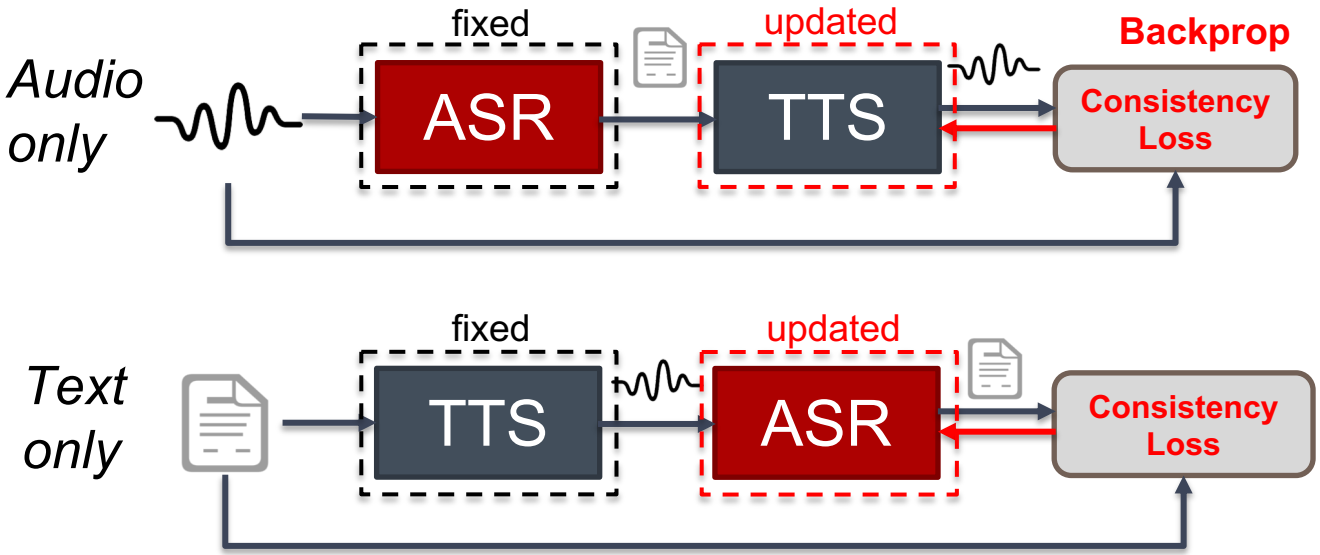


2. Discrete bottleneck makes the network not fully differentiable
   e.g. $\texttt{one\_hot}(\texttt{argmax}(\cdot))$

3. TTS requires more computation and memory

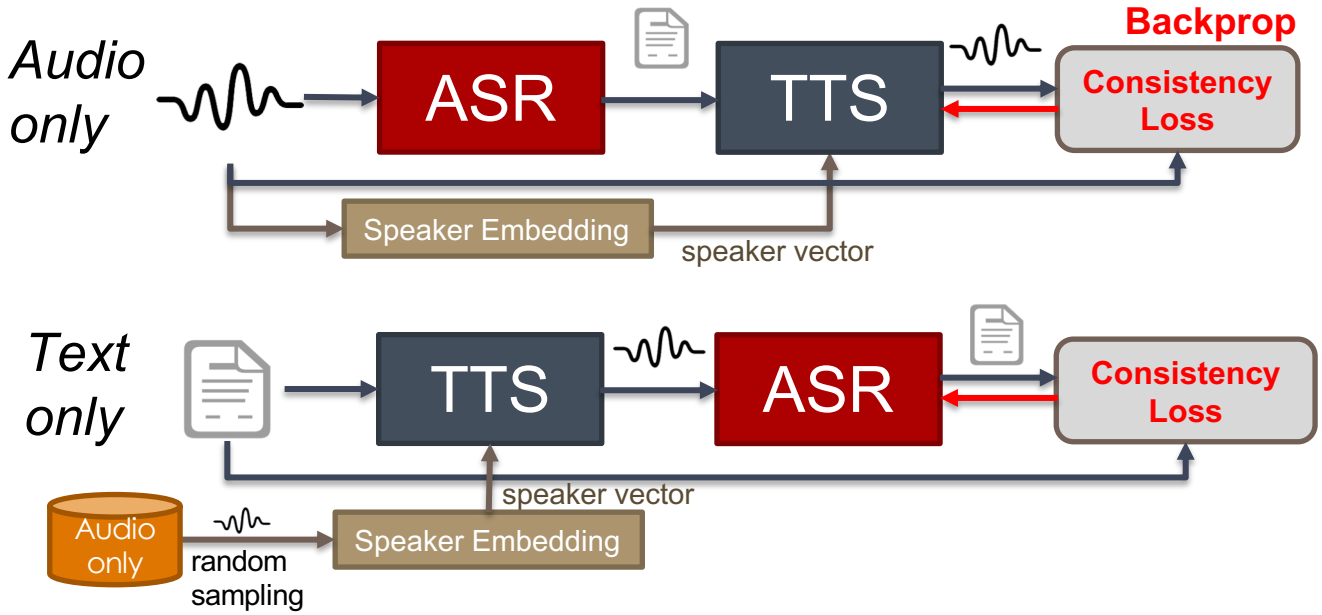
Speech chain model [Tjandra+ 2017]

- Update ASR and TTS alternately using unpaired data
- Avoid the discrete bottleneck by preventing backprop from TTS to ASR
- Also do not backprop from ASR to TTS

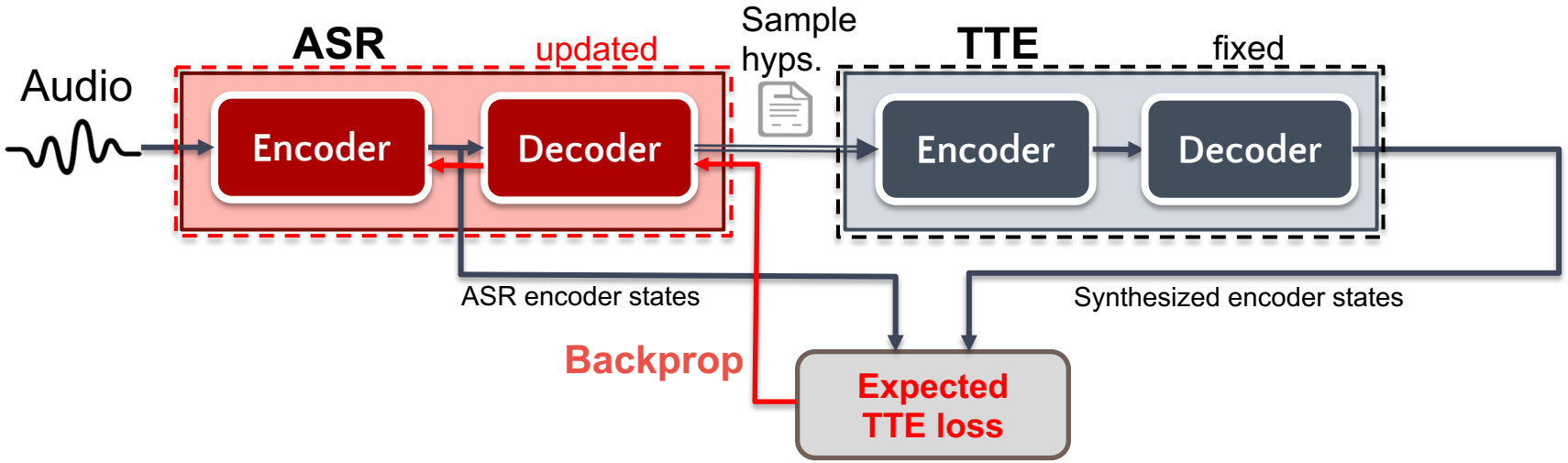


Speech chain model with speaker embedding [Tjandra+'18]

- Neural TTS (Tacotron2) can generate various speaker's voice using speaker embedding vectors [Jia+ 2018]
- Speaker embedding helps reduce speaker mismatches in the loss



TTE-based cycle-consistency training [Hori+'19]



- Use text-to-encoder states (TTE) model [Hayashi+ 2018] that generates ASR encoder states instead of mel-spectrum from text
- Avoid discrete bottleneck by reinforcement learning to backprop the expected TTE loss to ASR network
- Reduce speaker variability and computation cost

REINFORCE method to update the ASR model

- REINFORCE is a useful algorithm when the loss is not differentiable
- Use an expected loss w.r.t. character sequence $\mathbf{C}$ given audio input $\mathbf{X}$

$$\mathcal{L}_{\text{ette}} = \mathbb{E}_{\mathbf{C}|\mathbf{X}} \left[\mathcal{L}_{\text{tte}}(\hat{\mathbf{H}}^{\text{asr}}(\mathbf{C}), \mathbf{H}^{\text{asr}}(\mathbf{X})) \right]$$

$\mathcal{L}_{\text{tte}}(\cdot,\cdot)$: MSE between true and predicted state sequences

- Policy gradient technique to compute the gradients by sampling from ASR model

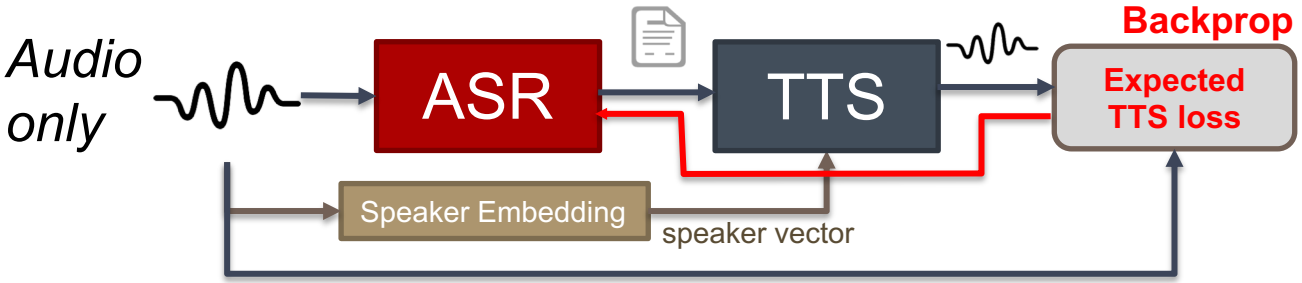
$$\nabla \mathcal{L}_{\text{ette}} \approx \frac{1}{N} \sum_{\substack{\mathbf{C}^n \sim p_{\text{asr}}(\cdot|\mathbf{X}), \\ n=1,\ldots,N}} \mathrm{T}(\mathbf{C}^n, \mathbf{X}) \nabla \log p_{\text{asr}}(\mathbf{C}^n|\mathbf{X})$$

$$\mathrm{T}(\mathbf{C}^n, \mathbf{X}) = \mathcal{L}_{\text{tte}}(\hat{\mathbf{H}}^{\text{asr}}(\mathbf{C}^n), \mathbf{H}^{\text{asr}}(\mathbf{X})) - B(\mathbf{X}, \mathbf{C}^n)$$

$B(\cdot,\cdot)$: baseline value to reduce the variance

Toward complete cycle-consistency training

- Expected TTS loss + speaker embedding (x-vector) [Baskar+ 2019]
  - More straightforward approach than TTE
  - Use unpaired audio and text for ASR-TTS and TTS-ASR cycles



- Straight-through estimator + Gumbel-Softmax [Tjandra+ 2019]
  - Gradients are passed directly to ASR based on approximation:

$$\frac{\partial}{\partial p_t}\texttt{one\_hot}(\texttt{argmax}(p_t)) \approx \mathbb{1}$$

p_t is obtained by `Softmax` or `GumbelSoftmax`

  - Not tested with semi-supervised training

Performance comparison in semi-supervised training for ASR [Baskar+ 2019]

LibriSpeech 100h (parallel) + 360h (unpaired)

Model	Unpaired data	RNNLM	%CER	%WER
Baseline	-	-	11.1	25.2
Backtranslation-TTE [Hayashi+ 2018]	Text	-	10.0	22.0
Cycle-TTS [Baskar+ 2019]	Text	-	8.0	17.9
Criticizing-LM [Liu+ 2019]	Text	yes	9.1	17.3
Cycle-TTS [Baskar+ 2019]	Text	yes	8.0	17.0
Cycle-TTE [Hori+ 2019]	Audio	yes	9.9	19.5
Cycle-TTS [Baskar+ 2019]	Audio	yes	7.8	16.8
Autoencoder [Karita+ 2019]	Both	yes	8.4	18.0
Cycle-TTS [Baskar+ 2019]	Both	-	7.6	17.5
Cycle-TTS [Baskar+ 2019]	Both	yes	**7.6**	**16.6**
Oracle (460h paired) [Hayashi+ 2018]	-	-	4.6	11.8

Approaching to the oracle result obtained with the same-sized paired data

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR)
b. Multispeaker speech recognition (SS + ASR)
c. Multichannel speech recognition (SE + ASR)
d. Speech translation (ASR + MT (+TTS))
e. Spoken language understanding (ASR + NLU)

(Integration 2) **Boost speech technologies**

f. Semi-supervised training of ASR models
   - Cycle-consistency training (ASR + TTS)

g. Other semi-supervised approaches

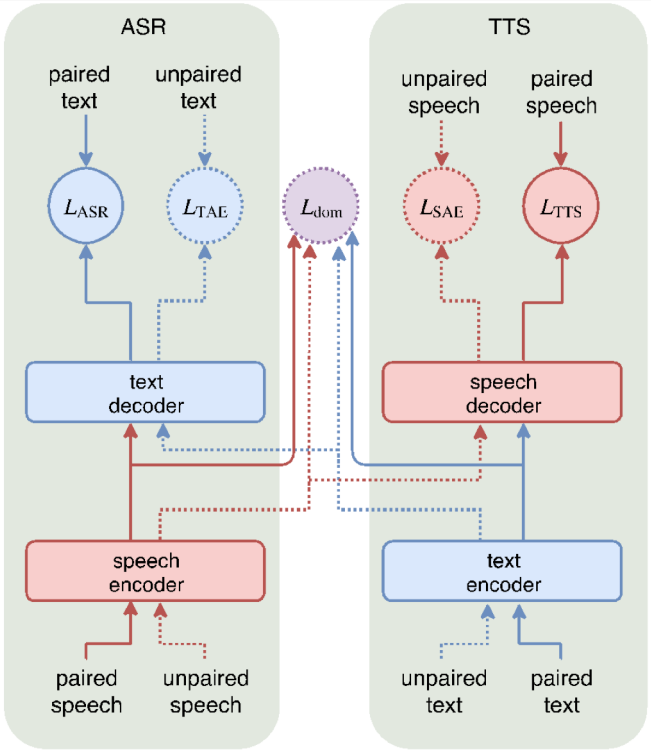
h. Domain adversarial training
   - Speaker-invariant training (ASR + SR)

ESPnet supports its recipe

g. Other semi-supervised approaches

Autoencoder approach [Karita+ 2019]

- Try to map speech and text to a common encoding space using an inter-domain loss:
  - Adversarial loss
  - Gaussian KL loss
  - MMD loss
- For paired data: ASR and TTS encoder decoders
- For unpaired audio input: Speech autoencoder
- For unpaired text input: Text autoencoder

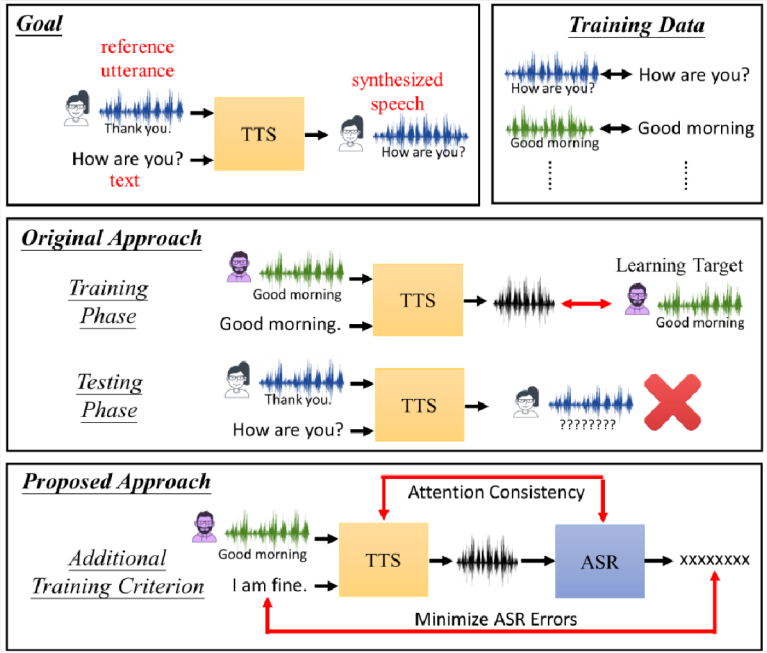


Karita, Shigeki, et al. "Semi-supervised End-to-end Speech Recognition Using Text-to-speech and Autoencoders." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

g. Other semi-supervised approaches

TTS with style transfer can be improved by ASR objective [Liu+ 2018]

- Problem when the reference audio and input text represent different sentences
- Introduce ASR loss, which is not affected by mismatches between reference audio and input text
- The authors also considered attention consistency between ASR and TTS



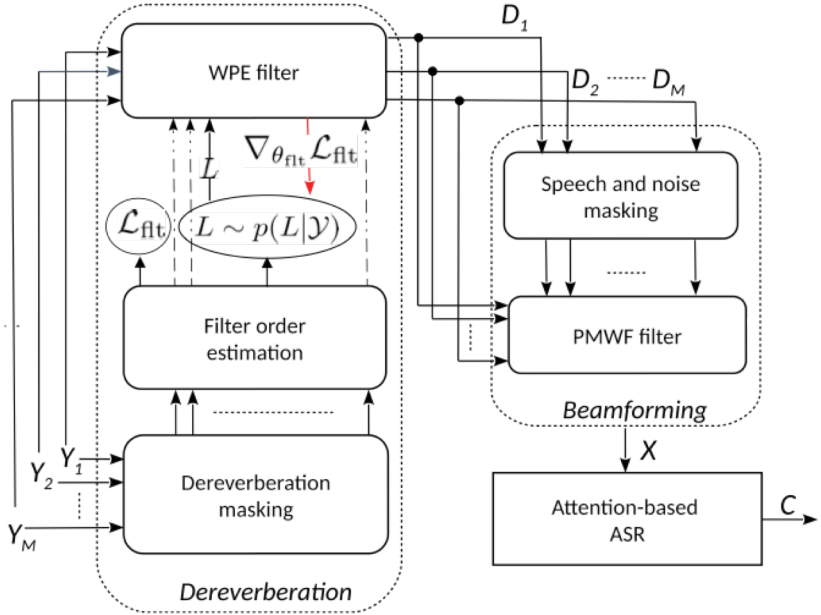
Liu, Da-Rong, et al. "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition." 2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018.

g. Other semi-supervised approaches

Multichannel speech enhancement using ASR objective

- Optimize BF with ASR objectives [Ochiai+ 2017]
- Optimize BF and dereverberation using ASR objective [Shanmugan+ 2019]
- Outperform conventional beamformers w.r.t. enhancement quality* when target clean speeches are not available (transcripts are needed)

*ESPnet supports 7 audio quality measures including PESQ, LLR, Cepstral distance, SRMR, and STOI.



Subramanian, Aswin Shanmugam et al., "Speech enhancement using end-to-end speech recognition objectives," 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). IEEE, 2019

2.4 Recent studies on neural end-to-end integration

(Integration 1) **Build integrated speech applications**

a. Multilingual speech recognition (LID + ASR)
b. Multispeaker speech recognition (SS + ASR)
c. Multichannel speech recognition (SE + ASR)
d. Speech translation (ASR + MT (+TTS))
e. Spoken language understanding (ASR + NLU)

- **Enhance speech technology**

f. Semi-supervised training of ASR models
   - Cycle-consistency training (ASR + TTS)

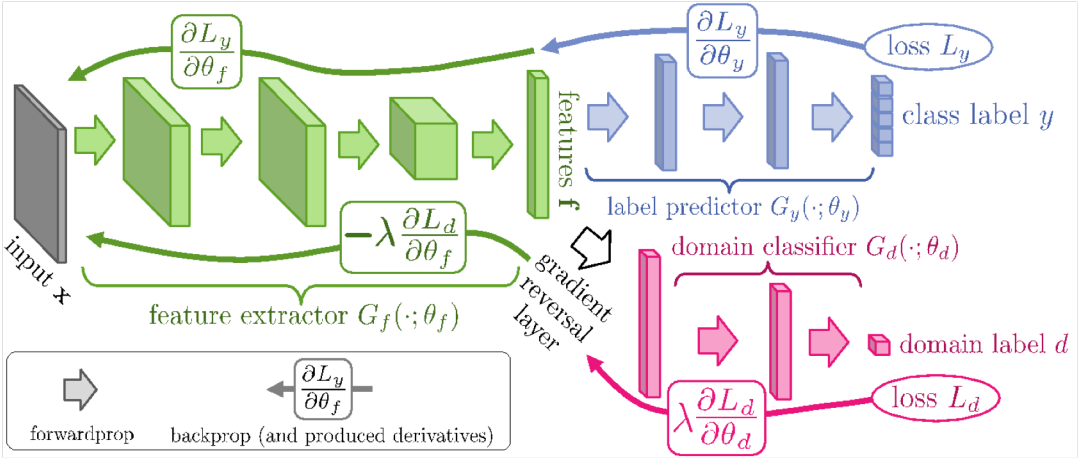
g. Other semi-supervised approaches

h. Domain adversarial training
   - **Speaker-invariant training (ASR + SR)**

ESPnet supports its recipe

h. Domain adversarial training

- Domain classifier helps learn domain independent models in an adversarial manner [Ganin+ 2016]

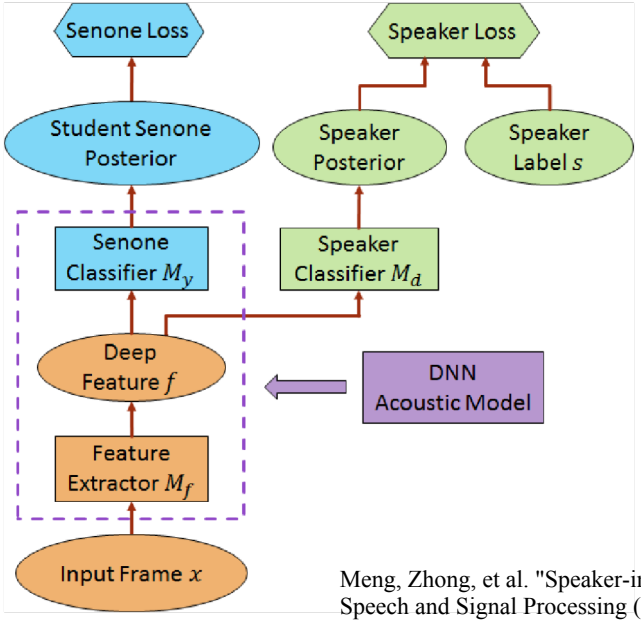


Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." The Journal of Machine Learning Research 17.1 (2016): 2096-2030.

- In ASR, acoustic models can be trained with a noise or speaker recognition network
  ⇨ The model becomes more robust to noise or speaker variations
  [Shinohara 2016, Saon+ 2017, Meng+ 2018]

Speaker-invariant training [Meng+'18]

- Jointly train a speaker classifier and a DNN acoustic model for ASR, so that the deep acoustic features (hidden vectors) fool the speaker classifier
- The model becomes more robust against speaker variations



System	Data	BUS	CAF	PED	STR	Avg.
SI	Real	24.77	16.12	13.39	17.27	17.84
	Simu	18.07	21.44	14.68	16.70	17.72
SIT	Real	22.91	15.63	12.77	16.66	**16.95**
	Simu	16.64	20.23	13.53	15.96	**16.54**

Table 1. The ASR WER (%) performance of SI and SIT DNN acoustic models on real and simulated development set of CHiME-3.

System	BUS	CAF	PED	STR	Avg.
SA SI	22.76	15.56	11.52	15.37	16.25
SA SIT	21.42	14.79	11.11	14.70	**15.46**

Table 2. The ASR WER (%) performance of SA SI and SA SIT DNN acoustic models after CRT unsupervised speaker adaptation on real development set of CHiME-3.

Meng, Zhong, et al. "Speaker-invariant training via adversarial learning." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

2.5 Summary of this section

- Neural end-to-end integration allows us to
  - Fully optimize the entire network to improve the performance for the target application
  - Design a special architecture to effectively train a network component with help of auxiliary networks
- There are many interesting studies on neural end-to-end integration
- Most of presented methods are included in ESPnet

3. End-to-End Speech Processing Toolkit (ESPnet)

Shinji Watanabe
Johns Hopkins University

ESPnet: End-to-end speech processing toolkit

Shinji Watanabe
Center for Language and Speech Processing
Johns Hopkins University

Joint work with Takaaki Hori , Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, Tsubasa Ochiai

at Interspeech 2018 and now over 60



All contributors (Github username, alphabet order)

@27jiangziyan, @Emrys365, @Fhrozen, @HawkAaron, @Hex-Lee, @JaejinCho, @Jzmo, @Magic-Bubble, @Masao-Someki, @Peidong-Wang, @ShigekiKarita, @SuperGops7, @TsubasaOchiai, @Xiaofei-Wang, @YosukeHiguchi, @akreal, @arendu, @b-flo, @bhigy, @bobchennan, @butsugiri, @creatorscan, @danoneata, @elgeish, @fanlu, @gtache, @hiratake55, @hirofumi0810, @indra622, @jan-schuchardt, @jheymann85, @jnishi, @kamo-naoyuki, @kan-bayashi, @karaf, @lumaku, @m-koichi, @m-wiesner, @mallidi, @marvinzh, @miguelvr, @mn5k, @naxingyu, @oadams, @potato-inoue, @pzelasko, @r9y9, @ruizhilijhu, @sas91, @shanguanma, @siddalmia, @simpleoier, @sknadig, @songmeixu, @sw005320, @synetkim, @takaaki-hori, @takenori-y, @unnonouno, @weiwchu, @ysk24ok, @zh794390558

Table of contents

Organization

- ESPnet GitHub Organization (https://github.com/espnet)
- Admin
  - Shinji Watanabe (JHU), Tomoki Hayashi (Nagoya Univ.), Shigeki Karita (NTT), Takaaki Hori (MERL)
- Developer
  - **Over 60 contributors** (https://github.com/espnet/espnet/graphs/contributors)
- CI, auto-documentation, version managements
- Initially designed to support the activity of **JSALT'18 workshop** "Multilingual End-to-end ASR for Incomplete Data" led by Hori and Watanabe
  - More than 20 researchers used ESPnet for this project
- Github star: > **1.3K**, **Over 60 citations** by Google Scholar

Frederick Jelinek Memorial Summer Workshop
(JSALT)

- 6-week workshop held every summer
- >10 researchers are gathered at the same place to work on a single topic
- **A lot of famous open source activities were born at the JSALT workshop**
  - SRILM
  - Kaldi
  - Moses

Organization

- ESPnet GitHub Organization (https://github.com/espnet)
- Admin
  - Shinji Watanabe (JHU), Tomoki Hayashi (Nagoya Univ.), Shigeki Karita (NTT), Takaaki Hori (MERL)
- Developer
  - **Over 60 contributors** (https://github.com/espnet/espnet/graphs/contributors)
- CI, auto-documentation, version managements
- Initially designed to support the activity of **JSALT'18 workshop** "Multilingual End-to-end ASR for Incomplete Data" led by Hori and Watanabe
  - More than 20 researchers used ESPnet for this project
- Github star: > **1.3K**, **Over 60 citations** by Google Scholar

ESPnet development policy

Make speech processing research more friendly

ESPnet, high-level overview

- **Open source (Apache2.0)** end-to-end speech processing toolkit
- **Chainer** or **PyTorch** based dynamic neural network toolkit
  - Easily develop novel neural network architecture
- Follows the famous speech recognition (**Kaldi**) style
  - Data processing, feature extraction/format
  - All-in-one recipes to provide a complete setup for speech processing experiments
- **Unified design** for various speech applications
- **Integration** of various speech applications (Section 2)
- **Pre-trained models** and **Notebook (on Google colab)**
- **State-of-the-art performance**

Code lines

- Kaldi

```
$ cat kaldi/src/*/*.{cc,cu,h} | wc –l
~330k
```

- ESPnet

```
$ wc -l `find ../../../../espnet_master/{espnet,utils}/
| grep -e "¥.sh" -e "¥.py"` | tail -n 1
~34k
```

  - Simple code thanks to Chainer/Pytorch as a main deep learning engine
  - Use Kaldi feature extraction, and python-based reader/writer (kaldiio)

Supported recipes (*38* recipes)

1. aishell
2. ami
3. an4 **(ASR/TTS test)**
4. aurora4
5. babel
6. chime4 **(multichannel ASR)**
7. chime5
8. cmu_wilderness **(multilingual ASR)**
9. commonvoice
10. csj
11. fisher_callhome_spanish **(speech translation)**
12. fisher_swbd
13. hkust
14. how2 **(speech/machine translation)**
15. hub4_spanish
16. iwsltT8 **(speech/machine translation)**
17. iwslt19 **(speech/machine translation)**
18. jnas
19. jsalt18e2e **(multilingual ASR)**
20. jsut
21. li10 **(multilingual ASR)**
22. libri_trans **(speech/machine translation)**
23. librispeech
24. libritts **(TTS)**
25. ljspeech **(TTS)**
26. m_ailabs **(TTS)**
27. mini_an4 **(ASR/TTS test)**
28. must_c **(speech/machine translation)**
29. reverb **(multichannel ASR)**
30. ru_open_stt
31. swbd
32. tedlium2
33. tedlium3
34. timit
35. voxforge
36. wsj
37. wsj_mix **(multispeaker ASR)**
38. yesno

Table of contents

ASR functionalities (preprocessing)

- **Kaldi style data preprocessing**
  - fairly comparable to the performance obtained by Kaldi hybrid DNN systems
  - easily porting the Kaldi recipe to the ESPnet recipe
- **Other feature extraction supports**
  - Librosa (https://librosa.github.io/librosa/)
  - Own feature extraction including speech enhancement and dereverberation
    - Keeping a computational graph for joint optimization
- **Data augmentation**
  - Kaldi data augmentation (e.g., speed and gain perturbation)
  - Specaugment

ASR functionalities (network architecture)

- **Character/Subword based modeling (Byte Pair Encoding)**
- **Five sequence to sequence models**
  - **CTC**
    - Toolkit built-in CTC and WarpCTC
  - **Attention-based encoder-decoder**
    - Subsampled BLSTM and/or VGG-like encoder and location attention
  - **Joint CTC/attention (Joint C/A)**
    - Multitask learning
    - Joint decoding with label-synchronous Joint C/A decoding (solve monotonic alignment issues)
  - **RNN Transducer**
    - Original RNN-T and RNN-attention
  - **Transformer**
    - Joint decoding with label-synchronous joint CTC/transformer

ASR functionalities (inference)

- **Label synchronous beam search**
  - Simple beam search without batching (readability)
  - Batch beam search (batchfy across utterances * hypotheses)
  - CTC and endpoint detection to make search less heuristic
- **Use of language models**
  - Combination of RNNLM trained with external text data (shallow fusion)
  - Word-level decoding
- **Include NIST scoring toolkit**
  - Provide a complete setup including the objective evaluation

Supported 35 languages

Major English tasks (WSJ, Fisher+Switchboard, Librispeech)

Japanese (Corpus of Spontaneous Japanese)

Mandarin (HKUST CTS)

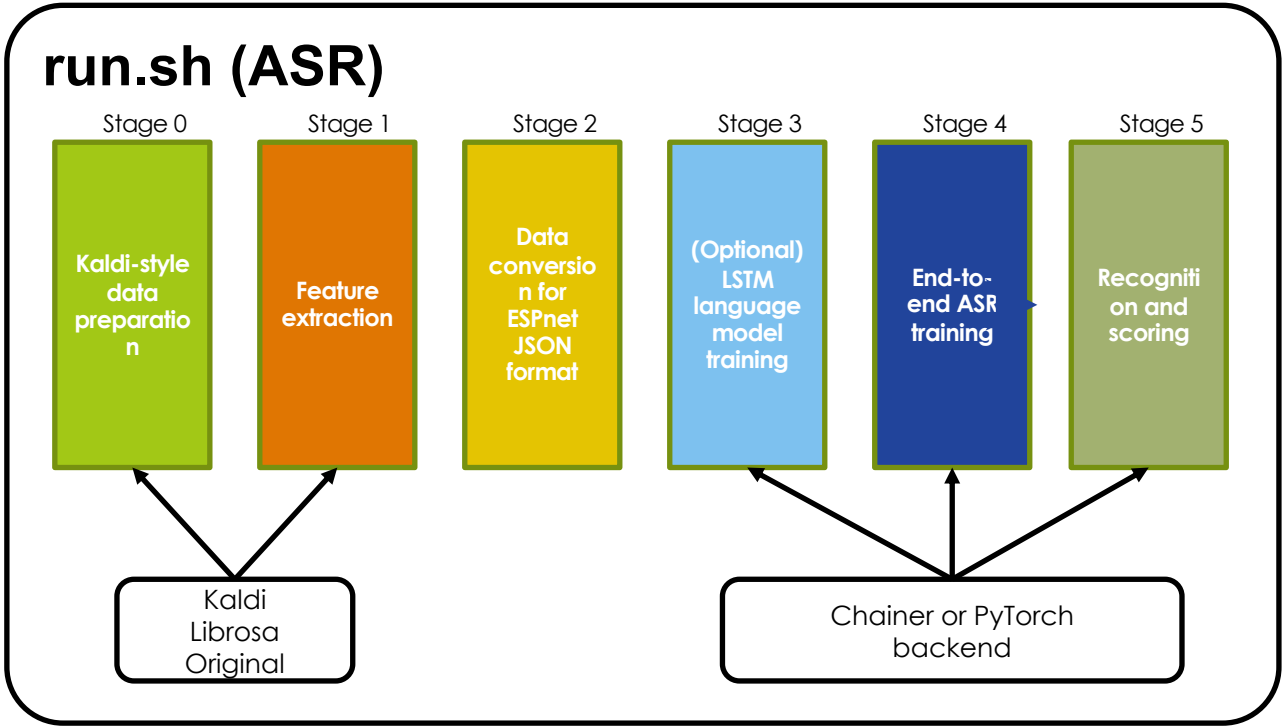
Babel 25 languages

- Cantonese , Assamese, Bengali, Pashto, Turkish, Georgian, Tagalog , Vietnamese , Haitian, Swahili, Lao, Tamil, Zulu, Kurmanji Kurdish, Tok Pisin

VoxForge 7 languages

- German, Spanish, French, Italian, Portuguese, Russian, Dutch

Basic flow of recipes



- **Very simple flow**
  - No Gaussian construction
  - No FST
  - No alignments
  - No lattice outpt
- **Easily ported from existing Kaldi recipes**
- **All-in-one recipe**
  - data download
  - data preparation
  - training & inference
  - scoring

TTS functionalities (preprocessing)

- **Kaldi style data preprocessing**
  - Almost the same as ASR recipe
  - Easily converted from ASR recipe (and vice versa)
- **Other feature extraction supports**
  - Librosa (https://librosa.github.io/librosa/)

TTS functionalities (network architecture)

- **Attention-based encoder-decoder**
  - Tacotron 2 [Shen+ 2017]
  - Multi-speaker extension w/ pre-trained spk-vector [Jia+ 2018]
- **Transformer**
  - TTS-Transformer [Li+ 2019]
  - Transformer + Tacotron 2's Post-net / Pre-net
  - Multi-speaker extension w/ pre-trained spk-vector
- **Feed-forward Transformer**
  - Fast Speech [Ren+ 2019]
  - Non-autoregressive feature generation

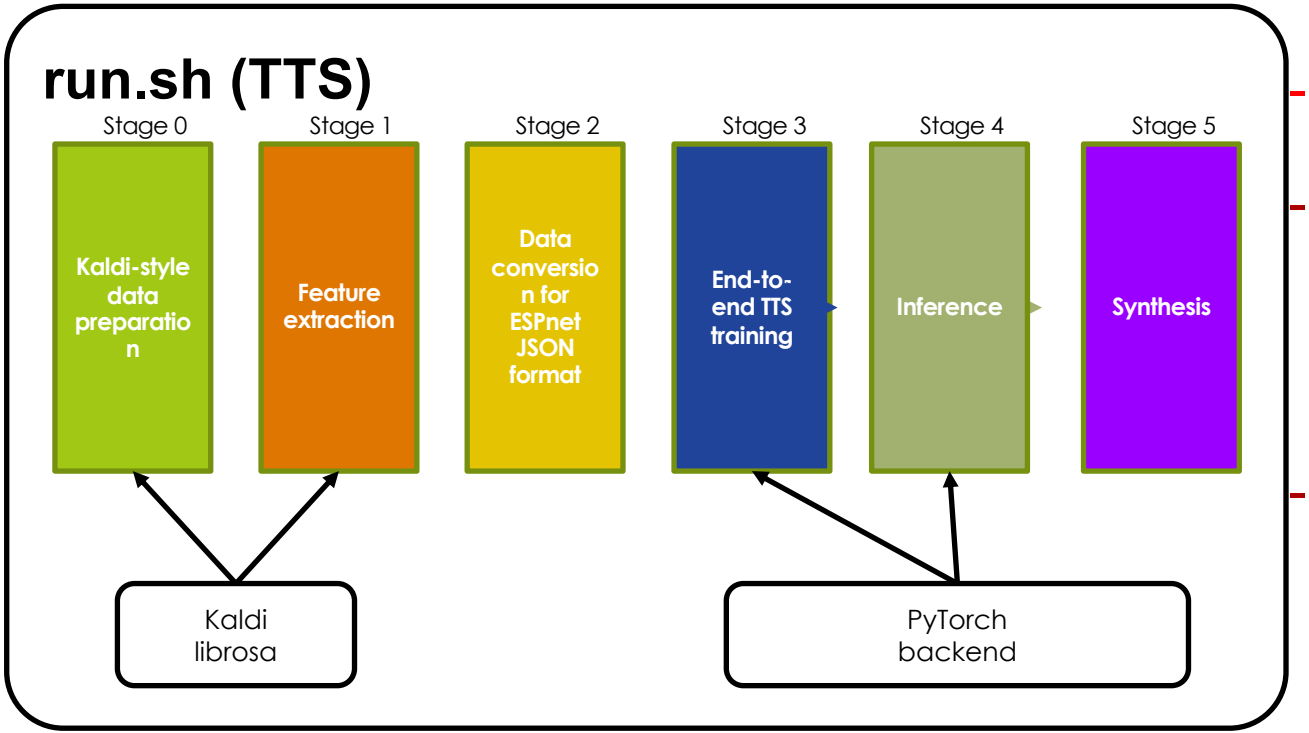
TTS functionalities (network related)

- **CBHG (1-D convolution bank + highway network + bidirectional GRU)**
  - From Tacotron [Wang+ 2017]
  - Conversion from Mel-filterbank to linear spectrogram
- **Additional attention function**
  - Forward attention with transition agent [Zhang+ 2018]
  - Forcing to be a causal attention
- **Additional loss function**
  - Guided attention loss [Tachibana+ 2018]
  - Forcing to be a diagonal attention

TTS functionalities (inference)

- **Speech feature generation**
  - RNN/Transformer based auto-regressive method
  - FastSpeech based non auto-regressive method
- **Waveform synthesis**
  - Griffin-Lim
  - Pre-trained Wavenet vocoder [Tamamori+ 2017] with noise shaping [Tachibana+ 2018]

Basic flow of recipes

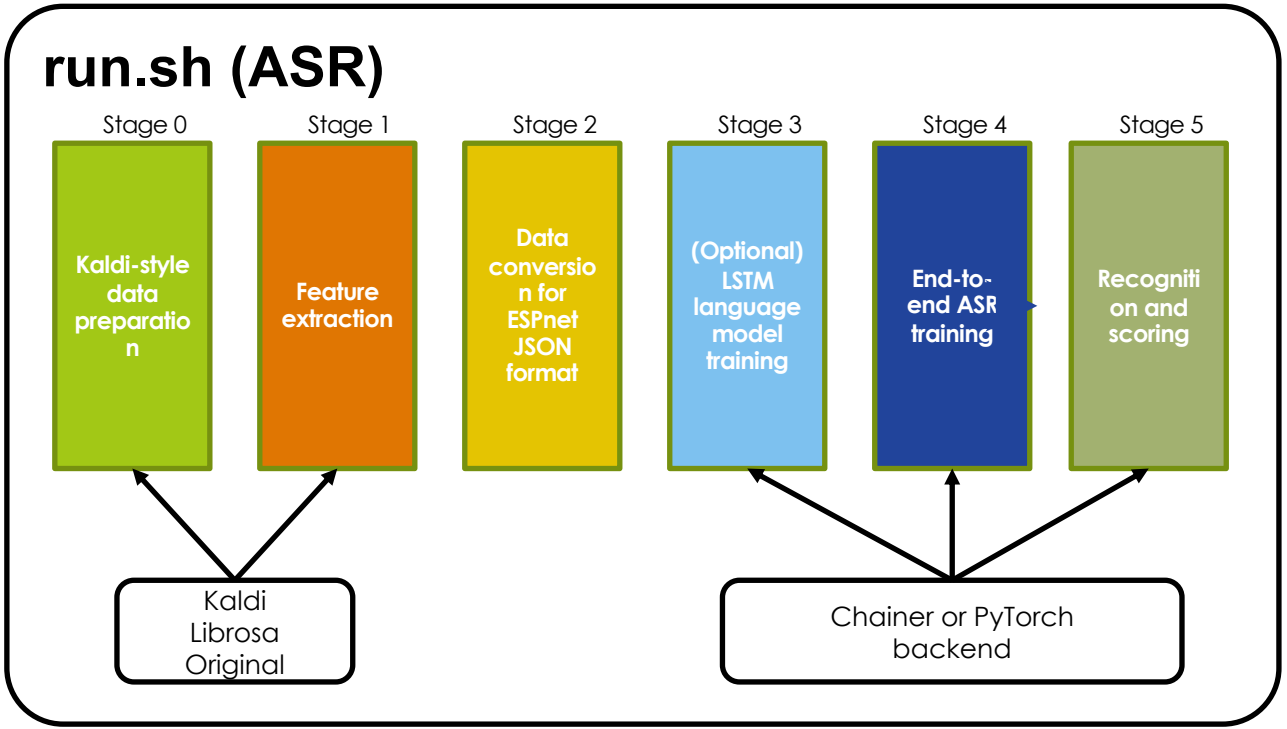


Quite similar flow to ASR one

This is a very unique open source tool to **support both ASR and TTS with same manners**

The next section provides more details about the flow

Basic flow of recipes



- **Very simple flow**
  - No Gaussian construction
  - No FST
  - No alignments
  - No lattice outpt
- **Easily ported from existing Kaldi recipes**
- **All-in-one recipe**
  - data download
  - data preparation
  - training & inference
  - scoring

Model Zoo

- We have been uploading various ASR/TTS models
  - https://github.com/espnet/espnet#asr-demo
  - https://github.com/espnet/espnet#tts-demo

Model	Notes
tedlium2.rnn.v1	Streaming decoding based on CTC-based VAD with uni-directional encoder-decoder
tedlium2.transformer.v1	Joint-CTC attention Transformer trained on Tedlium 2
tedlium3.transformer.v1	Joint-CTC attention Transformer trained on Tedlium 3
librispeech.transformer.v1	Joint-CTC attention Transformer trained on Librispeech
commonvoice.transformer.v1	Joint-CTC attention Transformer trained on CommonVoice

Model	Notes
libritts.tacotron2.v1	Multi-speaker Tacotron 2 with reduction factor = 2
ljspeech.tacotron2.v1	Tactoron 2 with reduction factor = 2
ljspeech.tacotron2.v2	Tacotron 2 with forward attention
ljspeech.tacotron2.v3	Tacotron 2 with guided attention loss
ljspeech.transformer.v1	Deep Transformer
ljspeech.transformer.v2	Shallow Transformer with reduction factor = 3
ljspeech.fastspeech.v1	Feed-forward Transformer with position-wise FFN
ljspeech.fastspeech.v2	Feed-forward Transformer with CNN instead of position-wise FFN
libritts.transformer.v1 (New!)	Multi-speaker Transformer with reduction factor = 2

TTS samples

- Automatically generate TTS sample webpage (see https://espnet.github.io/espnet-tts-sample/)

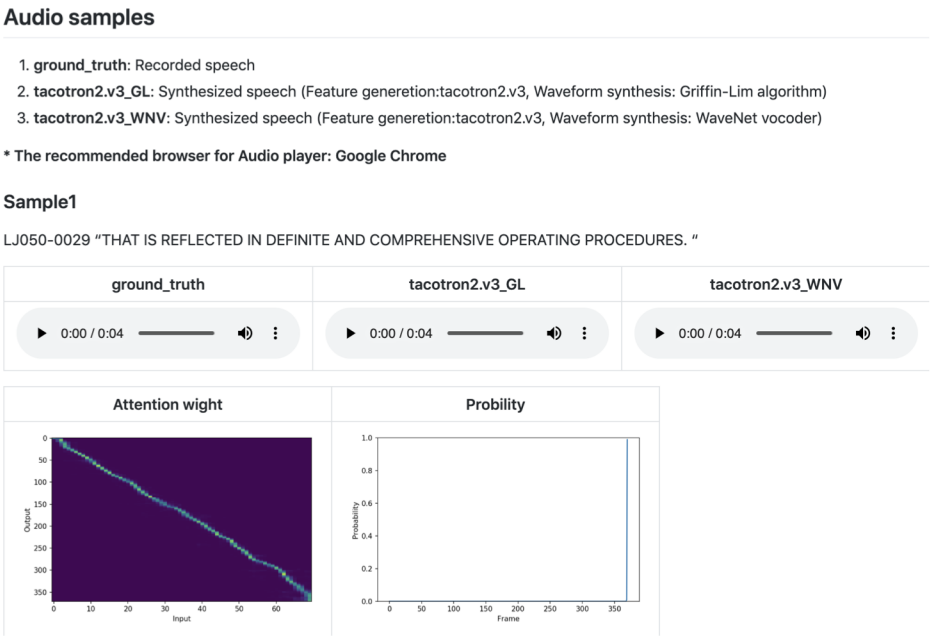


Table of contents

Experiments (< 100 hours, Nov 2018)

- Word Error Rate [%] in **English** Wall Street Journal (WSJ) task

Models	dev93	eval92	
ESPnet	7.0	**4.7**	Our best end-to-end
Attention model + word 3-gram LM [Bahdanau 2016]	-	9.3	
CTC + word 3-gram LM [Graves 2014]	-	8.2	
CTC + word 3-gram LM [Miao 2015]	-	7.3	
Attention model + word 3-gram LM [Chorowski 2016]	9.7	6.7	
Joint CTC/attention, multi-level LM	-	**5.6**	End-to-end best
Wav2Letter with gated convnet	-	**5.6**	
HMM/DNN + sMBR + word 3-gram LM	6.4	3.6	DNN/HMM (pipeline) best
HMM/DNN + sMBR + word RNN-LM	5.6	**2.6**	

Experiments (> 100 hours, Nov 2018)

- Character Error Rate [%] in HKUST **Mandarin** telephony task

Models	dev	
ESPnet	27.4	Our best end-to-end
CTC with language model [Miao (2016)]	34.8	End-to-end best
HMM/DNN + sMBR	35.9	
HMM/LSTM (speed perturb.)	33.5	
HMM/DNN + Lattice-free MMI (latest)	23.7	DNN/HMM (pipeline) best

- The gap comes from latest **sequence-discriminative** training progress
  → Full search to consider all possible decoding hypotheses

Performance boost by transformer [Karita+'19]

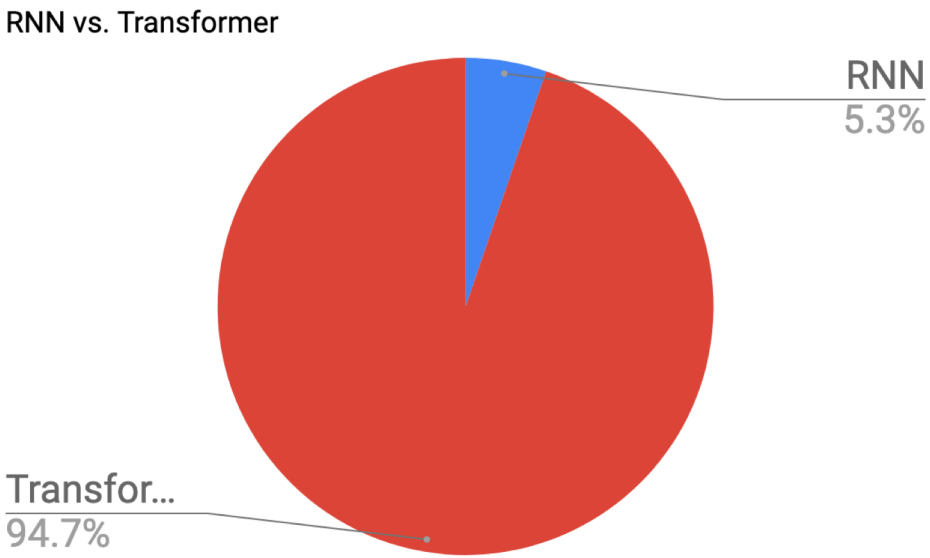
- ASR performance was improved in 13/15 tasks
- Reaching comparable performance to Kaldi

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

Performance boost by transformer [Karita+'19]

- ASR performance was improved in 13/15 tasks
- Reaching comparable performance to Kaldi

dataset	Our Transformer
AISHELL	**6.0 / 6.7**
AURORA4	**3.3 / 6.0 / 4.5 / 10.6**
CSJ	**5.7 / 4.1 / 4.5**
CHiME4	9.6 / 8.2 / 15.7 / 14.5
CHiME5	60.2 / 87.1
Fisher-CALLHOME Spanish	**27.0 / 26.3 / 24.4 / 45.3 / 46.2**
HKUST	**23.5**
JSUT	**18.7**
LibriSpeech	**2.2 / 5.6 / 2.6 / 5.7**
REVERB	**15.5 / 19.0**
SWITCHBOARD	26.0 / 14.0
TED-LIUM2	9.3 / **8.1**
TED-LIUM3	9.7 / 8.0
VoxForge	**9.4 / 9.1**
WSJ	6.8 / 4.4



Experiments (> 100 hours, Now)

- Character Error Rate [%] in HKUST **Mandarin** telephony task

Models	dev	
ESPnet	27.4	Our best end-to-end
Latest ESPnet	**23.5**	
CTC with language model [Miao (2016)]	34.8	
HMM/DNN + sMBR	35.9	
HMM/LSTM (speed perturb.)	33.5	
HMM/DNN + Lattice-free MMI (latest)	23.7	DNN/HMM (pipeline) best

- Transformer make end-to-end comparable or superior to DNN/HMM

Experiments (> 1000 hours, Now)

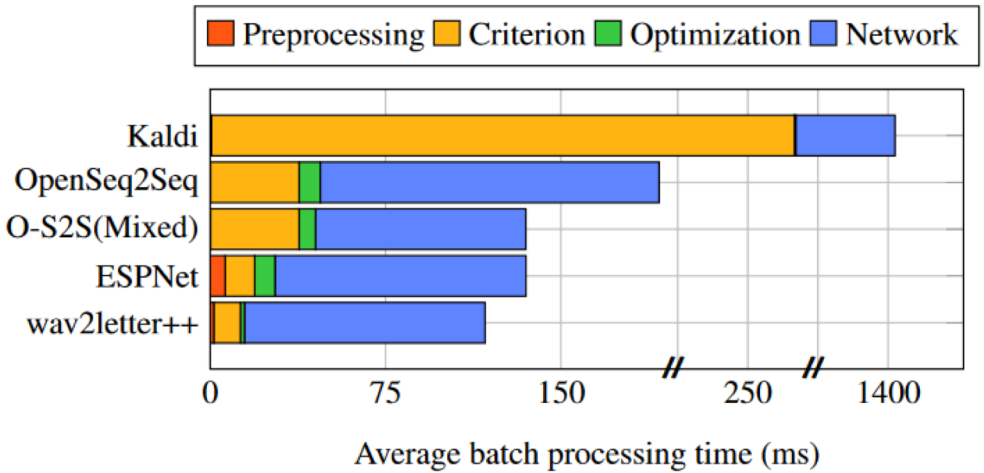
- Word Error Rate [%] in **English** Librispeech task

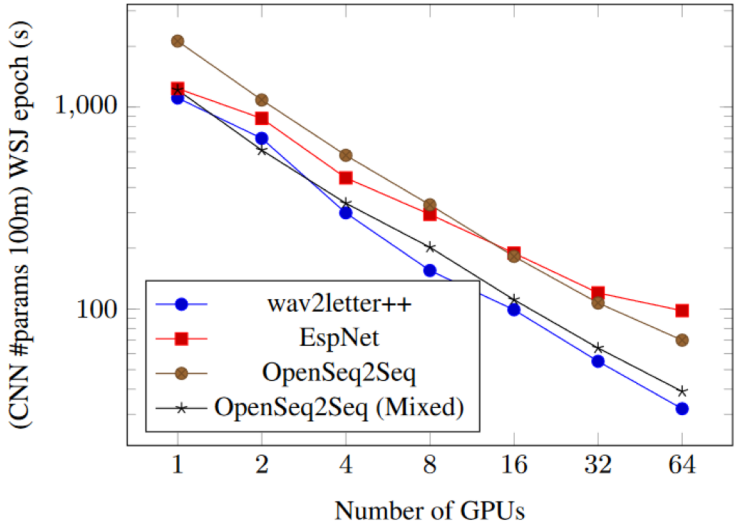
Methods	dev_clean	dev_other	test_clean	test_other
ESPnet	**2.2**	**5.6**	2.6	**5.7**
Facebook wav2letter++	3.1	10.1	3.4	11.2
RWTH E2E	2.9	8.8	3.1	9.8
Nvidia Jasper	2.6	7.6	2.8	7.8
Google SpecAug.	N/A	N/A	**2.5**	5.8

- **Reached Google's best performance by community-driven efforts**

Training Speed

ESPnet training speed is comparable to C++ toolkit: wav2letter
but ESPnet default decoding implementation is 10x slower





V. Pratap *et al.*, "Wav2Letter++: A Fast Open-source Speech Recognition System," *ICASSP 2019*, pp. 6460-6464. doi: 10.1109/ICASSP.2019.8683535

Decoding Speed

Vectorized implementation in ESPnet accelerates decoding in CPU/GPU [Seki+ 2019]

Effect of speed for LibriSpeech. Speed is measured in RTF on the test-clean set. In vectorized CTC-attention decoding ('+CTC'), a frame windowing technique is applied for fast computation of CTC prefix scores.

	ATT	+RNNLM	+CTC
Baseline (CPU)	1.13	1.14	1.24
Vectorized (CPU)	0.62	0.64	0.70
Vectorized (GPU)	0.03	0.03	0.09

Summary of this section

- ESPnet
  - Simple/unified software design
  - Reasonable and reproducible performance
  - ESPnet provides whole experimental procedure
  - Comparable ASR performance to the HMM/DNN (when >100h)
- The following section introduces ESPnet with more hands-on tutorials

4. Building End-to-End ASR and TTS Systems

Tomoki Hayashi
Nagoya University
Shigeki Karita
NTT Communication
Science Laboratories

Get the hands-on materials

Access from https://bit.ly/2kJQwgN

You need

- Internet connection
- Google Account
- Chrome browser (Recommended)

5. Conclusion and Future Research Directions

Shinji Watanabe
Johns Hopkins University

Conclusions

- Seq2seq concept provides a **unified** view for various speech processing
  ⚒ Unified software design
- End-to-end approaches can **integrate** various speech processing with a single neural network
  ⚒ Jointly optimize whole speech processing network
- ESPnet **implements** various speech processing applications based on these unification and integration
  ⚒ Accelerate your speech research with reasonable and reproducible performance

Future research directions

- Higher performance in each module (especially for ASR and TTS)
  - Need to fill out the gap from the conventional system
- End-to-end human conversation modeling
  - Integrate all speech processing applications
- Multimodal extension to integrate visual, language, and knowledge processing
  - e.g. Audio-visually-grounded spoken language applications [Sanabria+ 2018, Alamri+ 2019]
- Un-/semi-supervised training using other modalities

ESPnet development plans

- More documents, more examples, more models
- Fast/scalable training and inference
- Speech Embedding
- SLU Task
- Transformer LM (e.g. BERT, XLNet)
- Style Embedding (for TTS)
- Light-weight neural vocoder integration
  (e.g. WaveRNN, FFTNet)
- Fully customizable network architecture
  (e.g. Tacotron 2 encoder + Transformer decoder)

Please feel free to contact us if you have any requests/comments/suggestions at Github or questionary in https://forms.gle/xuwAakfmR3nhyv7E8

ESPnet development plans

- Check recent development plans in
  Issues: https://github.com/espnet/espnet/issues/1164
  PRs: https://github.com/espnet/espnet/pulls
  Milestones: https://github.com/espnet/espnet/milestones

- **Please feel free to contact us if you have any requests/comments/suggestions for the ESPnet development plans**
- **Also, we always welcome any contributions!**

Thanks a lot!

References

[Alamri+ 2019] H. Alamri, C. Hori, T. K. Marks, D. Batr, and D. Parikh, “Audio Visual Scene-aware dialog (AVSD) Track for Natural Language Generation in DSTC7,” in DSTC7 workshop at AAAI.

[Bahdanau+ 2016] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in ICASSP. IEEE, 2016, pp. 4945–4949.

[Baskar+ 2019] M. K. Baskar, S. Watanabe, R. Astudillo, T. Hori, L. Burget, and J. Černocký, “Self-supervised Sequence-to-sequence ASR using unpaired speech and text,” in Interspeech, 2019.

[Berard+ 2016] A. Berard, O. Pietquin, L. Besacier, and C. Servan, “Listen and translate: A proof of concept for end-to-end speech-to-text translation,” in NIPS Workshop, 2016.

[Chan+ 2016] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in ICASSP. IEEE, 2016, pp. 4960–4964.

[Chorowski+ 2015] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in Advances in Neural Information Processing Systems (NIPS), 2015, pp. 577–585.

[Dong+ 2018] L. Dong and S. Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in ICASSP. IEEE, 2018.

[Ganin+ 2016] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, 2016. Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 17(1), pp.2096-2030.

[Gibiansky+ 2017] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, “Deep voice 2: Multi-speaker neural text-to-speech,” in Advances in neural information processing systems, 2017, pp. 2962–2970.

References (cont'd)

[Gonzalez-Dominguez+ 2015] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. J. Moreno, "A real-time end-to-end multilingual speech recognition architecture," IEEE Journal of Selected Topics in Signal Processing, vol. 9, no. 4, pp. 749–759, 2015.

[Graves+ 2006] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in International Conference on Machine learning (ICML), 2006, pp. 369–376.

[Graves+ 2013] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in ICASSP. IEEE, 2013, pp. 6645–6649.

[Graves+ 2014] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in ICML, 2014, pp. 1764–1772.

[Haghani+ 2018] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters. "From Audio to Semantics: Approaches to end-to-end spoken language understanding." in IEEE Spoken Language Technology Workshop (SLT), pp. 720-726. IEEE, 2018.

[Hayashi+ 2017] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for wavenet vocoder," in ASRU. IEEE, 2017, pp. 712–718.

[Hayashi+ 2018] T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Multi-head decoder for end-to-end speech recognition," Interspeech, pp. 801–805, 2018.

[Hayashi+ 2018] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style data augmentation for end-to-end ASR," SLT, 2018.

References (cont'd)

[He+ 2016] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma, "Dual learning for machine translation," in Advances in Neural Information Processing Systems 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 820–828. 2016.

[Hori+ 2017] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in Interspeech, 2017.

[Hori+ 2019] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. L.Roux, "Cycle-consistency training for end-to-end speech recognition," in ICASSP. IEEE, 2019, pp. 6271-6275.

[Isik+ 2016] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single channel multi-speaker separation using deep clustering," in Interspeech, pp. 545–549.

[Jia+ 2018] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Lopez-Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis." In Advances in neural information processing systems, pp. 4480-4490. 2018.

[Jia+ 2019] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu. "Direct speech-to-speech translation with a sequence-to-sequence model." arXiv preprint, arXiv:1904.06037, 2019.

[Karita+ 2018] S. Karita, S. Watanabe, T. Iwata, A. Ogawa, and M. Delcroix, "Semi-supervised end-to-end speech recognition," Interspeech, pp. 2–6, 2018.

[Karita+ 2019] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders," in ICASSP. IEEE, 2019.

[Kim+ 2017] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in ICASSP. IEEE, 2017, pp. 4835–4839.

References (cont'd)

[Kim+ 2018] S. Kim and M. L. Seltzer, "Towards language-universal end-to-end speech recognition," in ICASSP. IEEE, 2018, pp. 4914–4918.

[Li+ 2019] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to human quality TTS with transformer," arXiv preprint arXiv:1809.08895, 2018.

[Liu+ 2018] D.-R. Liu, C.-Y. Yang, S.-L. Wu, and H.-Y. Lee, "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition," in IEEE Spoken Language Technology Workshop (SLT), 2018, pp. 640-647.

[Liu+ 2019] A. H. Liu, H.-Y. Lee, and L.-S. Lee, "Adversarial training of end-to-end speech recognition using a criticizing language model," in ICASSP. IEEE, 2019, pp. 6176-6180.

[Meng+ 2018] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gang, and B.-H. Juang, "Speaker-invariant training via adversarial learning," in ICASSP. IEEE, 2018, pp. 5969-5973.

[Miao+ 2015] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 167–174.

[Ochiai+ 2017] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1274–1288, 2017.

[Oliver+ 2019] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, "Massively Multilingual Adversarial Speech Recognition," in NAACL, 2019, pp. 96-108.

[Paszke+ 2017] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in NIPS-W, 2017.

[Povey+ 2011] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembeket al.,"The Kaldi speech recognition toolkit," in ASRU, 2011.

References (cont'd)

[Qian+ 2017] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," arXiv preprint, arXiv:1707.06527, 2017.

[Ren+ 2019] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "FastSpeech: Fast, robust and controllable text to speech," arXiv preprint arXiv:1905.09263, 2019.

[Sainath+ 2016] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, "Factored Spatial and Spectral Multichannel Raw Waveform CLDNNs," in ICASSP. IEEE, 2016.

[Sanabria+ 2018] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, "How2: a large-scale dataset for multimodal language understanding," arXiv preprint arXiv:1811.00347, 2018.

[Saon+ 2017] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L. L. Lim, and B. Roomi, "English Conversational Telephone Speech Recognition by Humans and Machines," in Interspeech 2017, pp.132-136.

[Seki+ 2018] H. Seki, S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," in ICASSP. IEEE, 2018, pp. 4919–4923.

[Seki+ 2018] H. Seki, T. Hori, S. Watanabe, J. Le Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in ACL, vol. 1, 2018, pp. 2620–2630.

[Seki+ 2019] H. Seki, T. Hori, S. Watanabe, N. Moritz, J. Le Roux, "Vectorized beam search for CTC-attention-based speech recognition," in Interspeech, 2019.

[Serdyuk+ 2018] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in ICASSP. IEEE, 2018, pp. 5754-5758.

References (cont'd)

[Settle+ 2018] S. Settle, J. Le Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in ICASSP, 2018, pp. 4819–4823.

[Subramanian+ 2019] A. S. Subramanian, X. Wang, S. Watanabe, T. Taniguchi, D. Tran, Y. Fujita, "An investigation of end-to-end multichannel speech recognition for reverberant and mismatch conditions, arXiv preprint, arXiv:1904.09049v3, 2019.

[Shen+ 2018] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang,Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in ICASSP. IEEE, 2018, pp. 4779–4783.

[Shinohara 2016] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition." in Interspeech, pp. 2369-2372. 2016.

[Sotelo+ 2017] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[Tachibana+ 2018] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in ICASSP, 2018, pp. 4784–4788.

[Tachibana+ 2018] K. Tachibana, T. Toda, Y. Shiga, and H. Kawai, "An investigation of noise shaping with perceptual weighting for wavenet-based speech generation," in ICASSP, 2018, pp. 5664–5668.

[Tamamori+ 2017] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent wavenet vocoder," Interspeech, pp. 1118–1122, 2017.

[Tjandra+ 2017] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking:Speech chain by deep learning," in ASRU, 2017, pp. 301–308.

[Tjandra+ 2018] A. Tjandra, S. Sakti, and S. Nakamura, "Machine Speech Chain with One-shot Speaker Adaptation." in Interspeech, 2018, pp. 887--891.

References (cont'd)

[Tjandra+ 2019] A. Tjandra, S. Sakti, and S. Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," in ICASSP. IEEE, 2019, pp. 6281-6285.

[Tokui+ 2015] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in Learning Sys in NIPS Workshop, vol. 5, 2015, pp. 1–6.

[Toshniwal+ 2018] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in ICASSP. IEEE, 2018, pp. 4904–4908.

[Vaswani+ 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N.Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in NIPS, 2017, pp. 5998–6008.

[Wang+ 2017] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio et al., "Tacotron: Towards end-to-end speech synthesis," in Interspeech, 2017, pp. 4006–4010.

[Wang+ 2018] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor,Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in ICML, 2018, pp. 5167–5176.

[Watanabe+ 2017a] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240–1253, 2017.

[Watanabe+ 2017b] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in ASRU, 2017, pp. 265–271.

[Watanabe+ 2018] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen et al., "ESPnet:End-to-end speech processing toolkit," in Interspeech, 2018, pp. 2207–2211.

[Weiss+ 2017] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in Interspeech, 2017, pp. 2625–2629.

References (cont'd)

[Xiao+ 2016] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. R. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in ICASSP. IEEE, 2016, pp. 5745–5749.

[Zhang+ 2018] J. Zhang, Z. Ling, and L. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in ICASSP, 2018, pp. 4789–4793.

[Zhu+ 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks." In 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242-2251. IEEE, 2017.