

Unifying **Speech Recognition** and **Generation** with Machine Speech Chain

Andros Tjandra, Sakriani Sakti, Satoshi Nakamura

Nara Institute of Science & Technology, Nara, Japan
RIKEN AIP, Japan

Outline

- Motivation
- Machine Speech Chain
- Sequence-to-Sequence ASR
- Sequence-to-Sequence TTS
- Experimental Setup & Results
- Conclusion

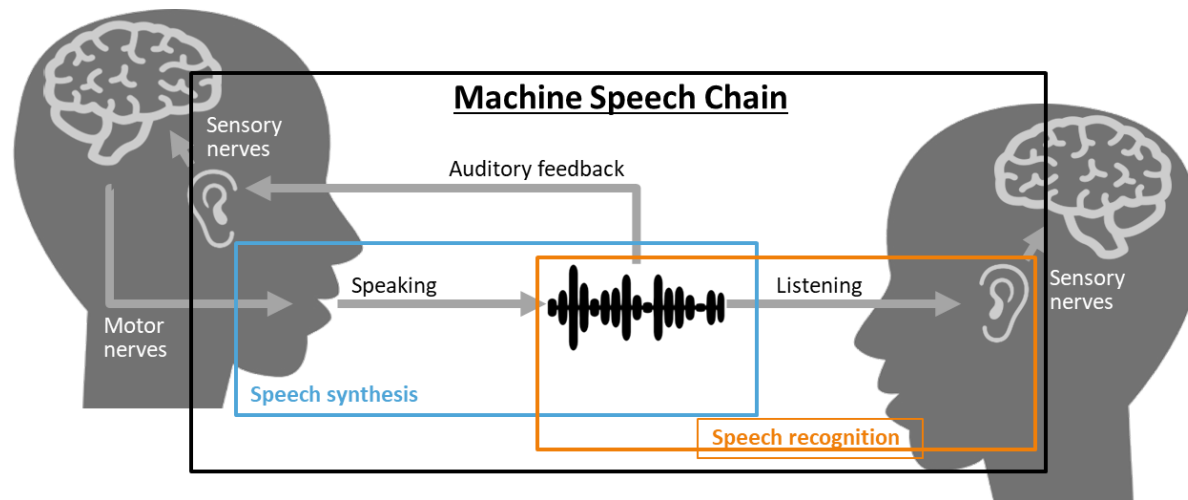
Motivation

- ASR and TTS researches have progressed independently without exerting much mutual influence on each other.

Property	ASR	TTS
Speech features	MFCC Mel-fbank	MGC log F0, Voice/Unvoice, BAP
Text features	Phoneme Character	Phoneme + POS + LEX (full context label)
Model	GMM-HMM Hybrid DNN/HMM End-to-end ASR	GMM-HSMM DNN-HSMM End-to-end TTS

Motivation (2)

- In human communication, a closed-loop speech chain mechanism has a critical auditory feedback mechanism.

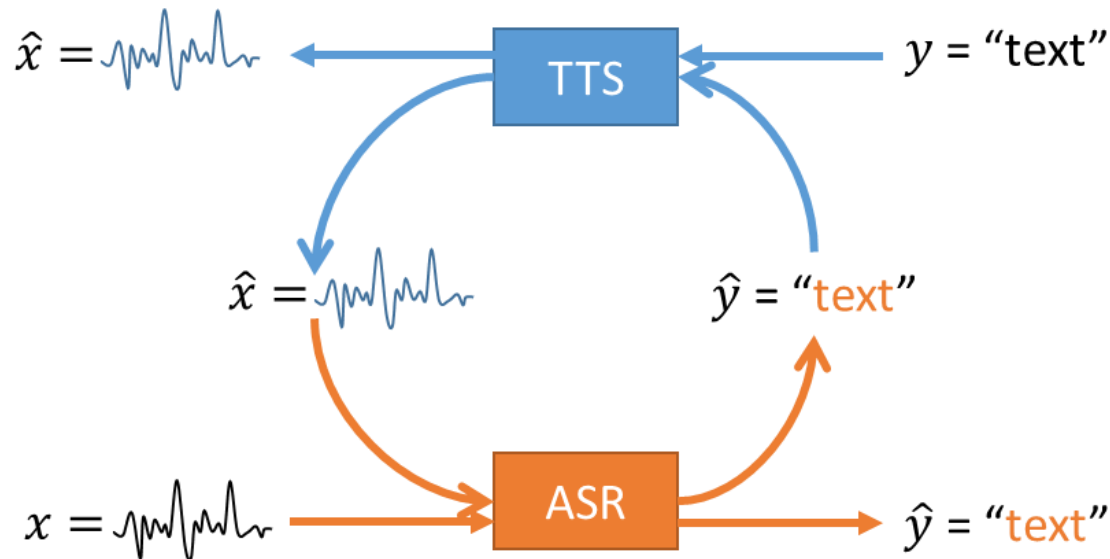


- Children who lose their hearing often have difficulty to produce clear speech.

This paper proposed ...

- Develop a closed-loop speech chain model based on deep learning model
- The benefit of closed-loop architecture :
 - Train both ASR & TTS model together
 - Allow us to concatenate both labeled and unlabeled speech & text (semi-supervised learning)
 - In the inference stage, we could use both ASR & TTS module independently

Machine Speech Chain

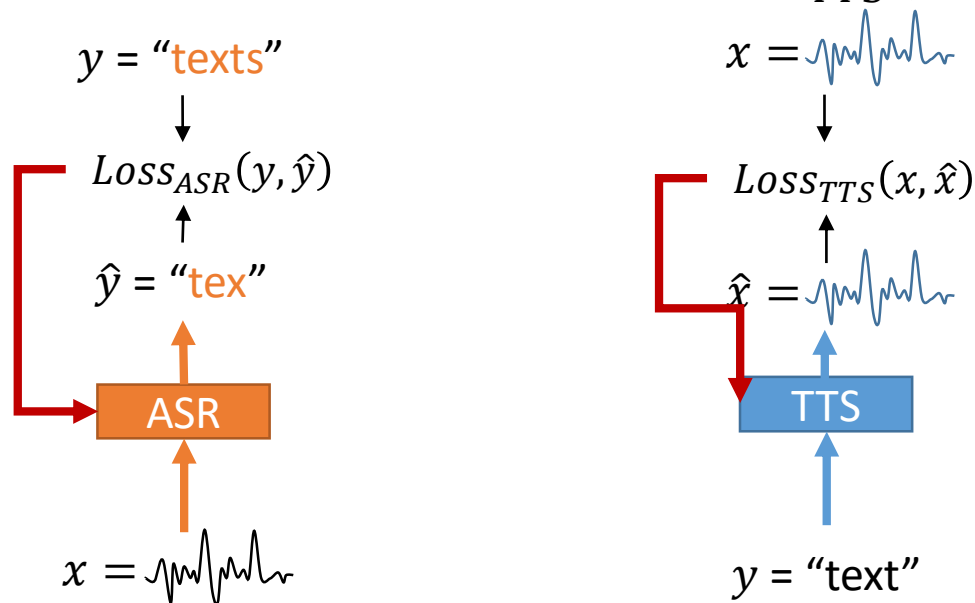


- Definition:

- x = original speech, y = original text
- \hat{x} = predicted speech, \hat{y} = predicted text
- $ASR(x): x \rightarrow \hat{y}$ (seq2seq model transform speech to text)
- $TTS(y): y \rightarrow \hat{x}$ (seq2seq model transform text to speech)

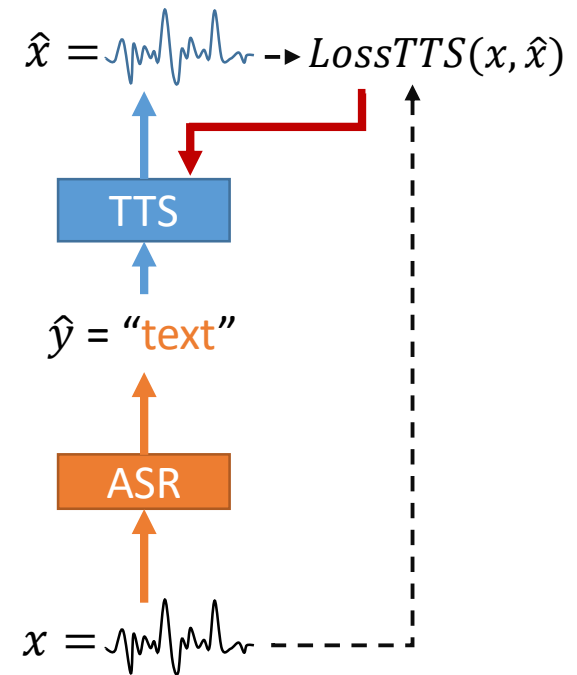
Machine Speech Chain (2)

- Case #1: Supervised training
 - We have a pair speech-text (x, y)
 - Therefore we could directly optimized *ASR* by minimize $Loss_{ASR}(y, \hat{y})$
 - and *TTS* by minimizing loss between $Loss_{TTS}(x, \hat{x})$



Machine Speech Chain (2)

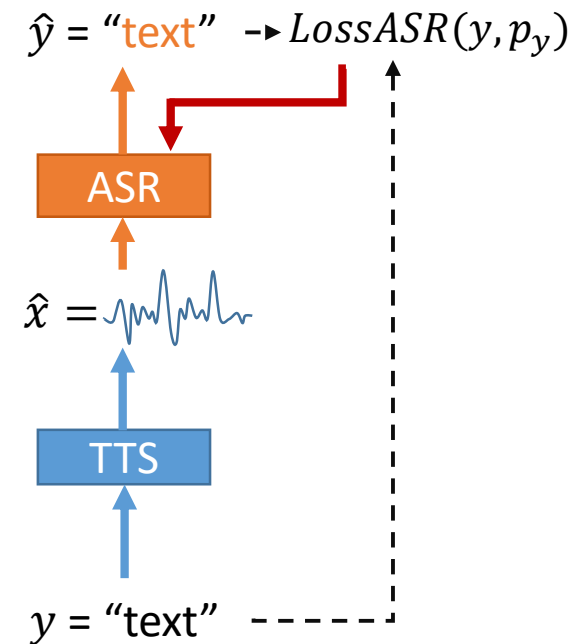
- Case #2: Unsupervised training with speech only
 1. Given the unlabeled speech features x
 2. ASR predicts most possible transcription \hat{y}
 3. TTS based on \hat{y} tries to reconstruct speech features \hat{x}
 4. Calculate $Loss_{TTS}(x, \hat{x})$ between original speech features x and predicted \hat{x}



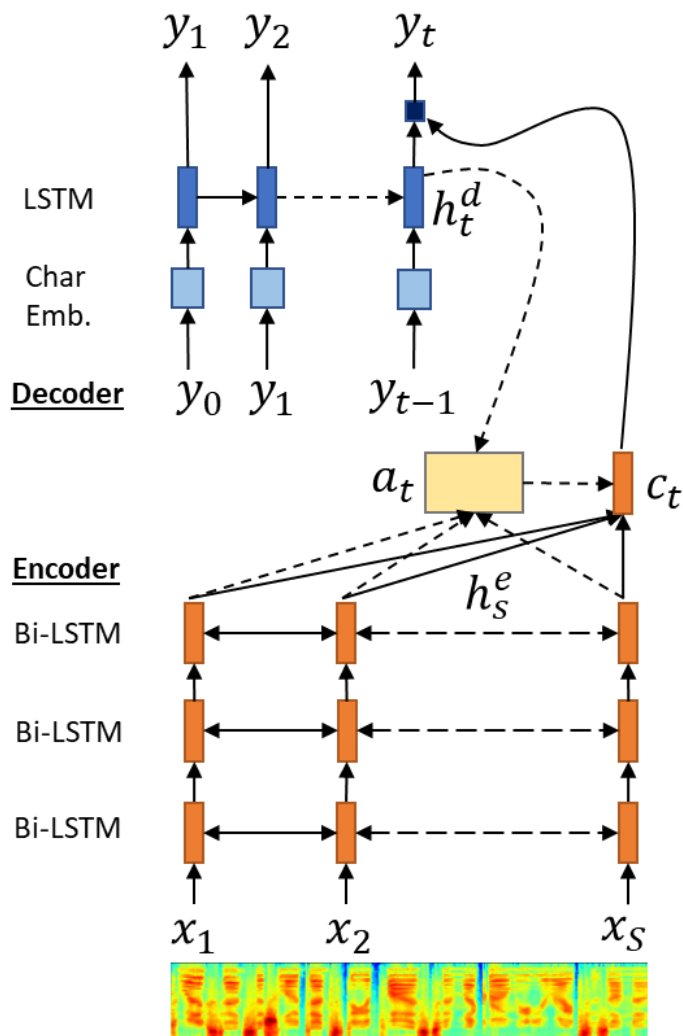
Machine Speech Chain (2)

- Case #3: Unsupervised training with text only

1. Given the unlabeled text features y
2. TTS generates speech features \hat{x}
3. ASR given \hat{x} tries to reconstruct speech features \hat{y}
4. Calculate $Loss_{ASR}(y, \hat{y})$ between original text y and predicted \hat{y}



Sequence-to-Sequence ASR



Input & output

- $x = [x_1, \dots, x_S]$ (speech feature)
- $y = [y_1, \dots, y_T]$ (text)

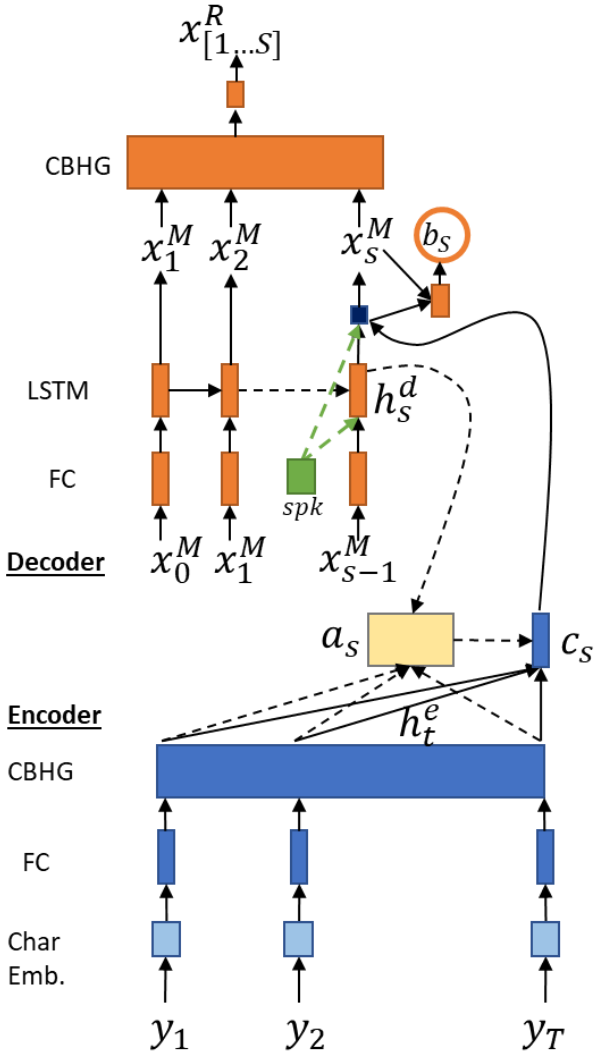
Model states

- $h_{[1..S]}^e$ = encoder states
- h_t^d = decoder state at time t
- a_t = attention probability at time t
 - $a_t(s) = \text{Align}(h_s^e, h_t^d)$
 - $a_t(s) = \frac{\exp(\text{Score}(h_s^e, h_t^d))}{\sum_{s=1}^S \exp(\text{Score}(h_s^e, h_t^d))}$
- $c_t = \sum_{s=1}^S a_t(s) * h_s^e$ (expected context)

Loss function

$$\mathcal{L}_{ASR}(y, p_y) = -\frac{1}{T} \sum_{t=1}^T \sum_{c \in [1..C]} 1(y_t = c) * \log p_{y_t}[c]$$

Sequence-to-Sequence TTS



Input & output

- $x^R = [x_1, \dots, x_S]$ (linear spectrogram feature)
- $x^M = [x_1, \dots, x_S]$ (mel spectrogram feature)
- $y = [y_1, \dots, y_T]$ (text)

Model states

- $h_{[1...S]}^e$ = encoder states
- h_s^d = decoder state at time t
- a_s = attention probability at time t
- $c_s = \sum_{t=1}^s a_s(t) * h_t^e$ (expected context)

Loss function

$$\mathcal{L}_{TTS1}(x, \hat{x}) = \frac{1}{S} \sum_{s=1}^S (x_s^M - \hat{x}_s^M)^2 + (x_s^R - \hat{x}_s^R)^2$$

$$\mathcal{L}_{TTS2}(b, \hat{b}) = -\frac{1}{S} \sum_{s=1}^S (b_s \log(\hat{b}_s) + (1 - b_s) \log(1 - \hat{b}_s))$$

$$\mathcal{L}_{TTS}(x, \hat{x}, b, \hat{b}) = \mathcal{L}_{TTS1}(x, \hat{x}) + \mathcal{L}_{TTS2}(b, \hat{b})$$

Settings

- Features
 - Speech:
 - 80 Mel-spectrogram (used by ASR & TTS)
 - 1024-dim linear magnitude spectrogram (SFFT) (used by TTS)
 - TTS reconstruct speech waveform by using Griffin-Lim to predict the phase & inverse STFT
 - Text:
 - Character-based prediction
 - a-z (26 alphabet)
 - 6 punctuation mark (,: ' ? . -)
 - 3 special tags <s> </s> <spc> (start, end, space)

Experiment on Single-Speaker

- Dataset

- BTEC corpus (text), speech generated by Google TTS (using gTTS library)
- Supervised training: 10000 utts (text & speech paired)
- Unsupervised training: 40000 utts (text & speech unpaired)

- Result

Data	Hyperparameter			ASR	TTS		
	α	β	gen. mode	CER (%)	Mel	Raw	Acc (%)
Paired (10k)	-	-	-	10.06	7.07	9.38	97.7
+Unpaired (40k)	0.25	1	greedy	5.83	6.21	8.49	98.4
	0.5	1	greedy	5.75	6.25	8.42	98.4
	0.25	1	beam 5	5.44	6.24	8.44	98.3
	0.5	1	beam 5	5.77	6.20	8.44	98.3

Experiment on Multi-Speaker Task

- Dataset

- BTEC ATR-EDB corpus (text & speech) (25 male, 25 female)
- Supervised training: 80 utts / spk (text & speech paired)
- Unsupervised training: 360 utts / spk (text & speech unpaired)

- Result

Data	Hyperparameter			ASR	TTS		
	α	β	gen. mode	CER (%)	Mel	Raw	Acc (%)
Paired (80 utt/spk)	-	-	-	26.47	10.21	13.18	98.6
+Unpaired (remaining)	0.25	1	greedy	23.03	9.14	12.86	98.7
	0.5	1	greedy	20.91	9.31	12.88	98.6
	0.25	1	beam 5	22.55	9.36	12.77	98.6
	0.5	1	beam 5	19.99	9.20	12.84	98.6

Conclusion

- Proposed a speech chain based on deep-learning model
- Explored applications in single and multi-speaker tasks
- Results: improved ASR & TTS performance by teaching each other using only unpaired data
- Future work: Perform real-time feedback mechanisms similar to human approach

😊 Thank you for listening 😊