Towards Incremental ASR and TTS for Real-time Interaction

Satoshi Nakamura[1,2], with
Sakriani Sakti[1,2], Tomoya Yanagita[1], Novitasari Sashi[1]
[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Advanced Intelligence Project AIP, Japan

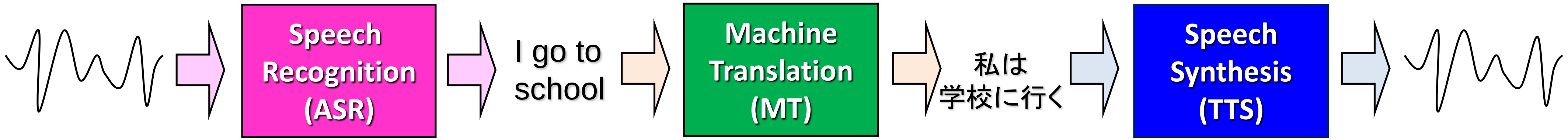
Topics

- Incremental Speech Processing for Real-time Interaction
  - Incremental ASR
  - Incremental TTS

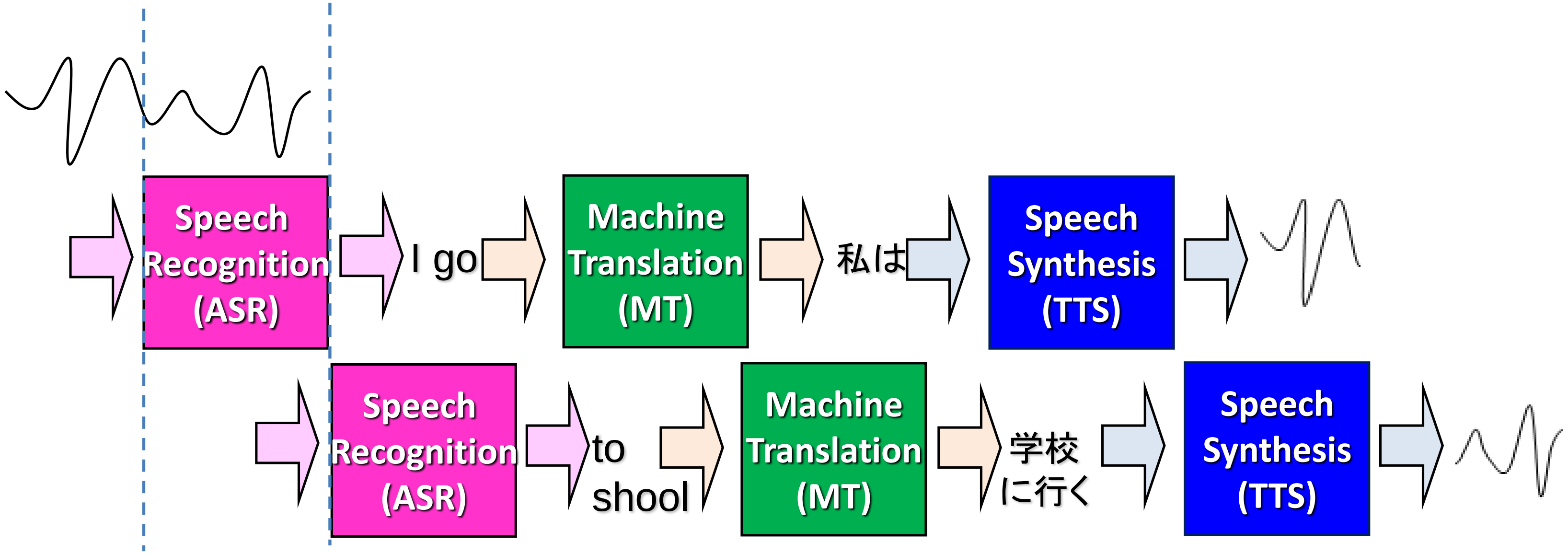
- Application
  - Simultaneous Speech Translation
  - Spoken Dialogue System

Real-time Machine Speech Interpreter

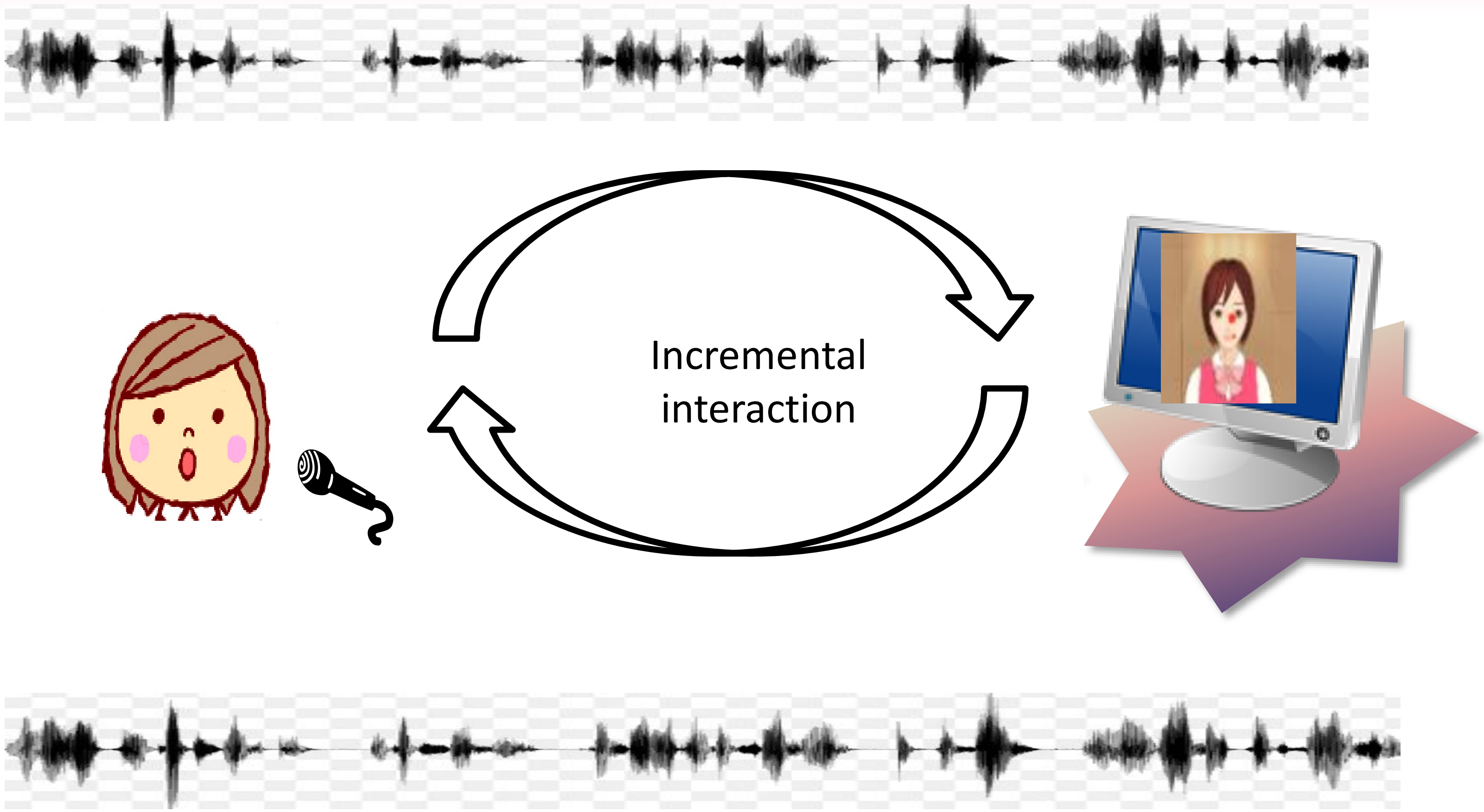
- Traditional Speech Translation



- Real-time Machine Speech Interpreter



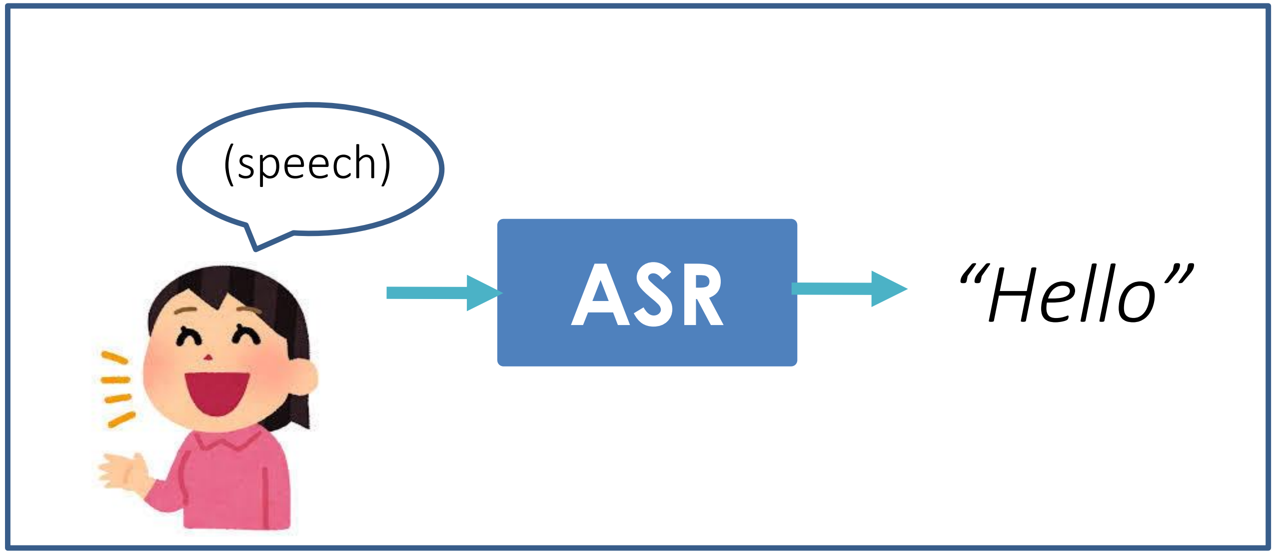
Dialogue System



Neural Incremental Speech Recognition

Automatic Speech Recognition System

- **ASR system** transcribes speech into text
- Task examples:
    - Spoken dialog system
    - Speech translation
    - Closed-caption generation, etc.



Neural Incremental Speech Recognition

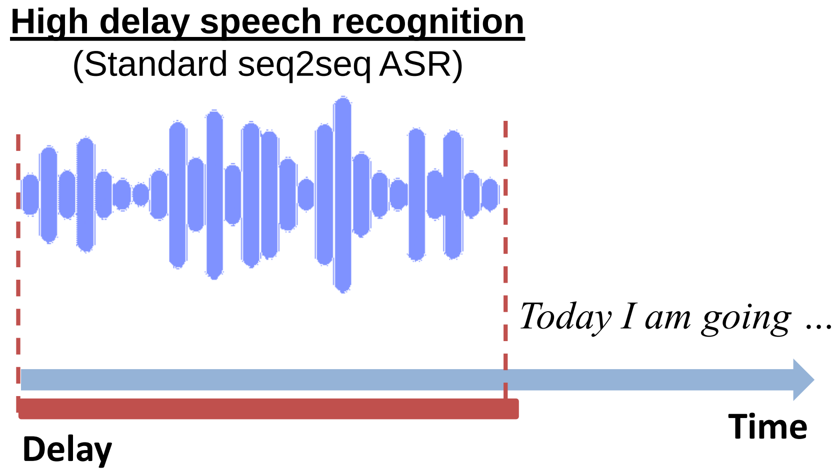
Automatic Speech Recognition System (2)

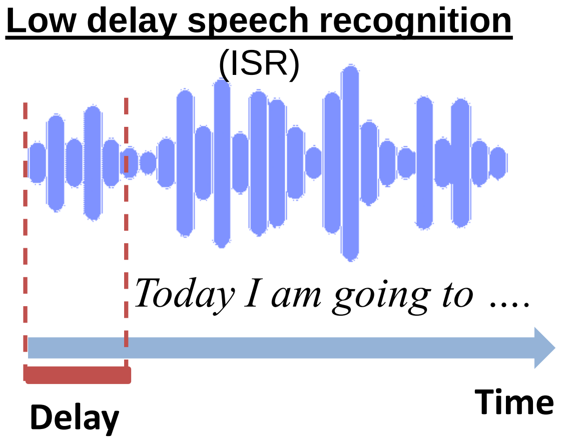
- ***State-of-the-art***

  Sequence-to-sequence neural ASR (end-to-end)
  - Standard encoder-decoder with a global attention mechanism
  - Output prediction starts after the input speech finish
    - → High accuracy but high delay

      e.g. a 5 minutes speech requires more than 5 minutes to be recognized
  - Unsuitable for real-time tasks
    - Real-time speech translation
    - Live video closed-caption generation
    - Real-time meeting transcription generation, etc.

- **Incremental ASR (ISR)** for low-delay speech recognition



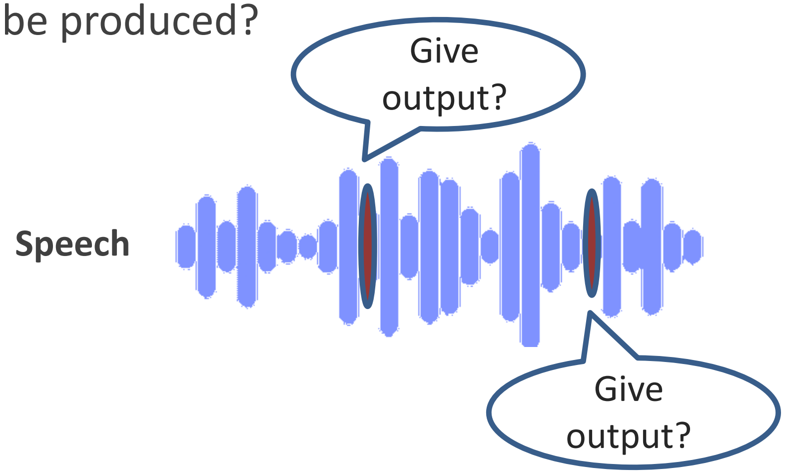


Incremental Speech Recognition

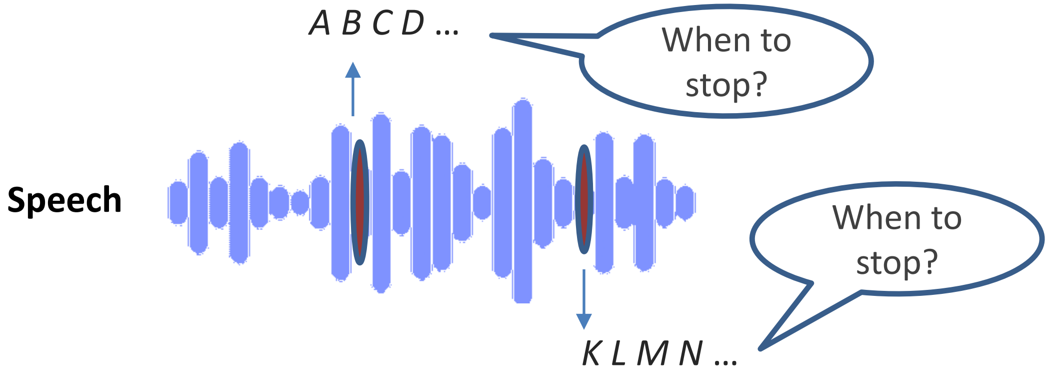
- **ISR** begins the speech recognition without waiting the speech to finish (low delay)
  - Recognize the speech part-by-part in several incremental steps
  - Input: a short part of the speech

- **Challenge:** How to do an incremental step?

1) Input boundary decision
When the transcription of a short speech part can be produced?



2) Output boundary decision
When to stop the output prediction of the current speech part and and move to the next?



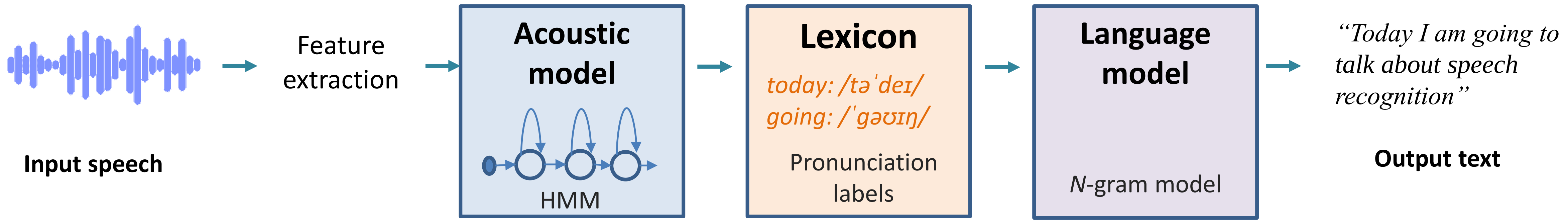
Need to learn short input-short output alignments

Incremental Speech Recognition

Related Works

A. Statistical approach (Pipeline)

- ❖ Hidden Markov model (HMM) ASR [Rabiner, 1989; Juang and Rabiner, 1991]
- ❖ 3 parts: Acoustic model, lexicon, language model



- ❖ Low delay speech recognition by performing left-to-right input processing (unidirectional)
- ❖ Not end-to-end

Neural Incremental Speech Recognition

How to achieve an ISR system that can:

1. reduce delay,
2. keep the system complexity, and
3. maintain a close performance

of the standard neural ASR system?

Proposal

Neural ISR construction by employing sources (architecture, knowledge) from standard neural ASR.

Neural Incremental Speech Recognition

Attention-Transfer Incremental Speech Recognition

Sashi Novitasari, Andros Tjandra, Sakriani Sakti, Satoshi Nakamura, "Sequence-to-sequence Learning via Attention Transfer for Incremental Speech Recognition", Interspeech 2019, Graz, Austria, DOI: 10.21437/Interspeech.2019-2985, 3835-3839, Sep. 2019

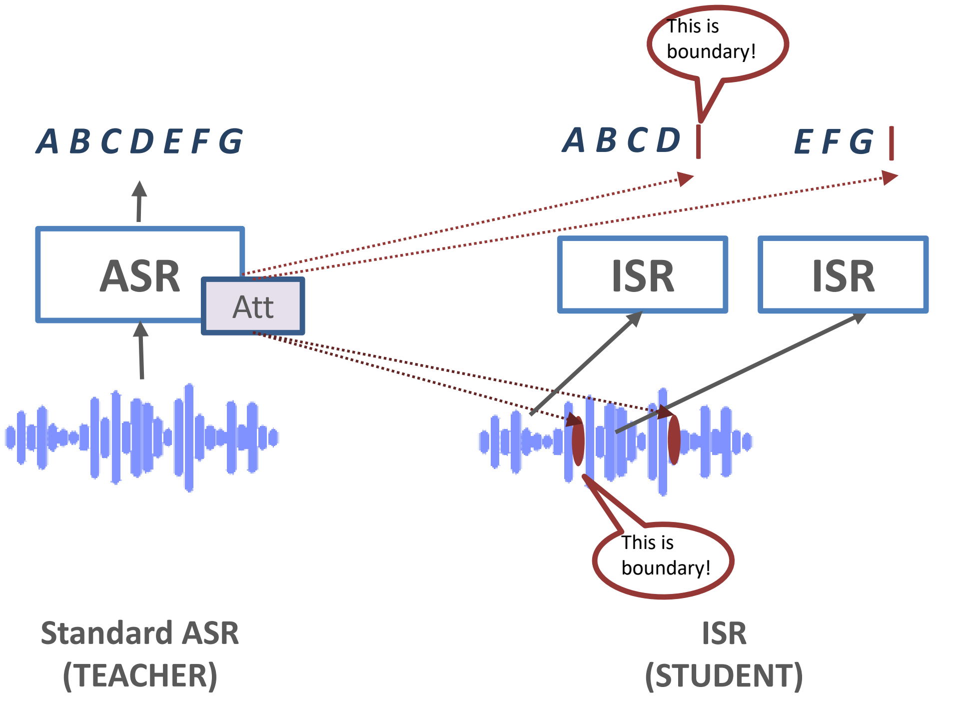
Attention-Transfer Incremental Speech Recognition (AT-ISR)

[Novitasari et al., 2019]

- **Aim**
  ISR (student) learns to mimic the attention-based alignment generated by a standard seq2seq ASR (teacher)
  - ISR architecture : Same as the teacher (seq2seq)
  - Incremental step : Learn through attention transfer from the teacher ASR

- **Attention transfer** : Attention knowledge transfer from teacher to student model
  - Prev. works → image recognition tasks
    - Teach another model [Zaguruyko and Komodakis, 2017]
    - Domain transfer (image to video) [Li et al., 2017]
  - Has not been utilized for ISR construction yet

AT-ISR Training



Neural Incremental Speech Recognition
Attention-Transfer Incremental Speech Recognition

Overview

Seq2seq ASR: Encoder-Decoder with Attention

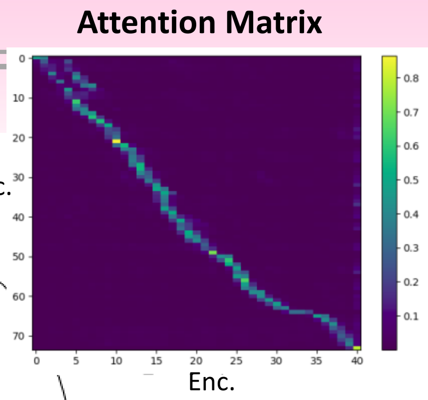
Output: Character (basic model)

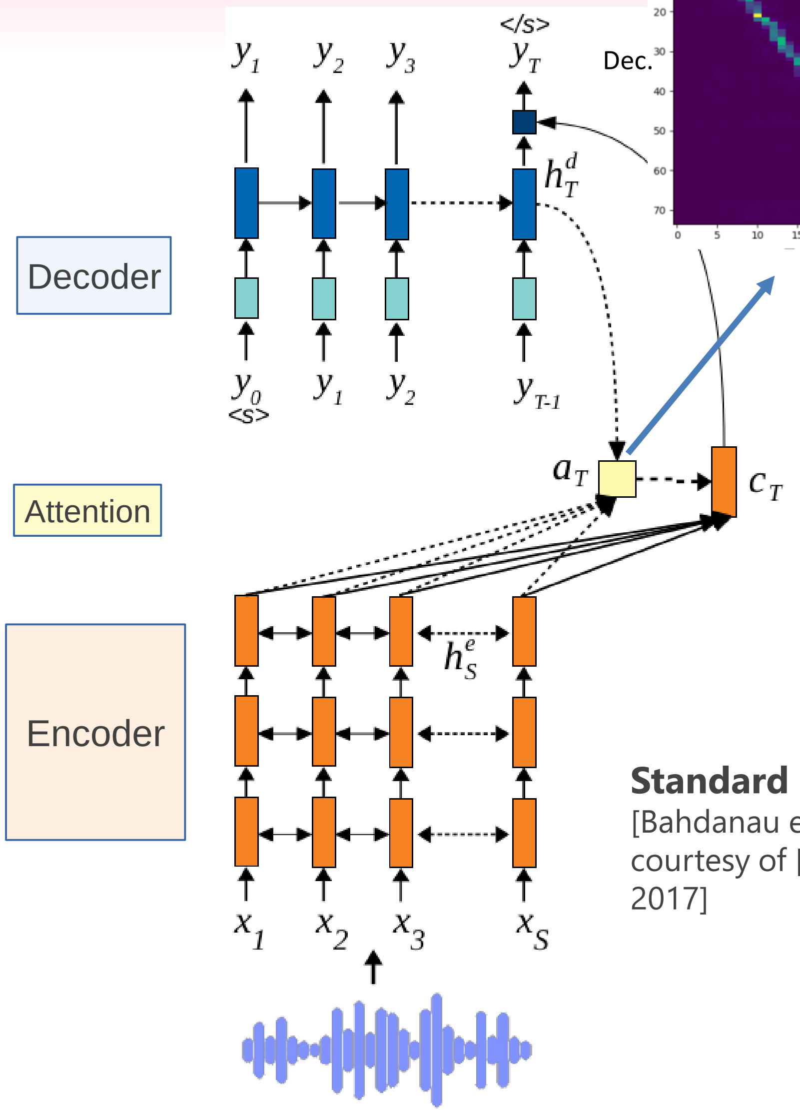
Components

- **Encoder (recurrent network)**
  Encode input features sequence ($\boldsymbol{X}$) into hidden states ($\boldsymbol{h}^e$)
- **Decoder (recurrent network)**
  Predict token sequence ($\boldsymbol{Y}$) based on input context information (c_t) and prev. text ($\boldsymbol{Y}_{<t}$) for each t-th token
- **Attention (linear network)**
  For each t-th token prediction, compute c_t based on alignment scores of $\boldsymbol{h}^e, h_t^d$

$$c_t = \sum_{s=1}^{S} a_t(s) * h_s^e \quad \Bigg| \quad a_t(s) = \frac{exp(Score(h_s^e, h_t^d))}{\sum_{s=1}^{S} exp(Score(h_s^e, h_t^d))}$$

s = Encoding timestep; t = Decoding timestep



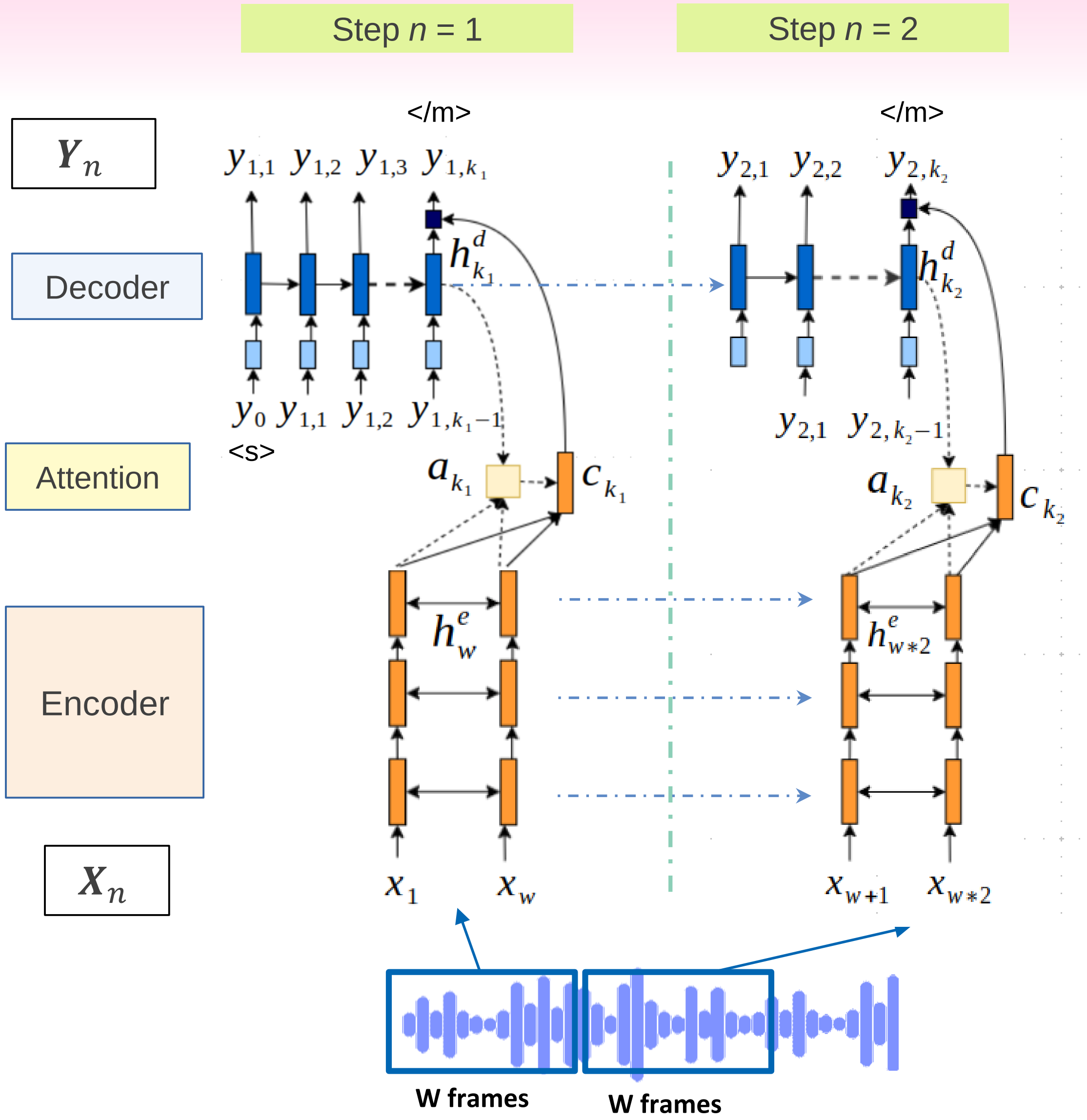


Standard Seq2seq ASR
[Bahdanau et al, 2015], courtesy of [Tjandra et al., 2017]

AT-ISR Recognition Method

- Given: Full speech ($\boldsymbol{X}$), length S
- Recognize the speech segment-by-segment sequentially based on a fix-sized input window
- For each incremental recognition step n:
  1. **Encode** $\boldsymbol{X}_n$, a W speech frames from $\boldsymbol{X}$ $(W < S)$
  2. **Decode** for $\boldsymbol{Y}_n$ that aligns with $\boldsymbol{X}_n$, until an *end-of-block* *</m>* token is predicted or max. length is reached
     - **Attend** the input $\boldsymbol{X}_n$
  3. **Shift** the input window W frames by keeping the model's state

  (Total step number: $N = \frac{S}{W}$)
- Incremental step:
  - Input boundary : last speech frame in the input window
  - Output boundary : *</m>* token in the output text
- Alignment learning → Attention transfer



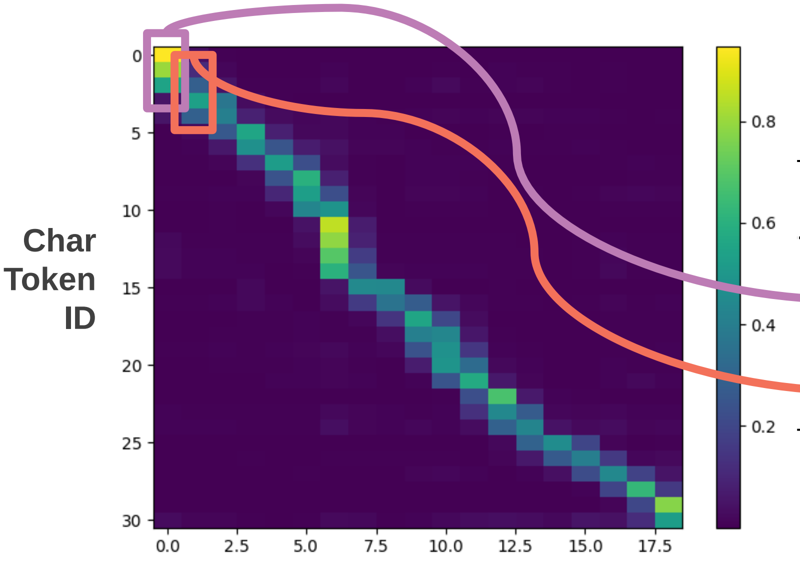
Attention-Transfer Incremental Speech Recognition

Attention Transfer

Train ISR (student) to learn the attention-based alignment from a standard seq2seqASR (teacher)

1) Extract speech-text alignment from attention matrix generated by the teacher ASR during teacher-forcing text generation (alignment pair = high attention score):

Teacher ASR attention matrix

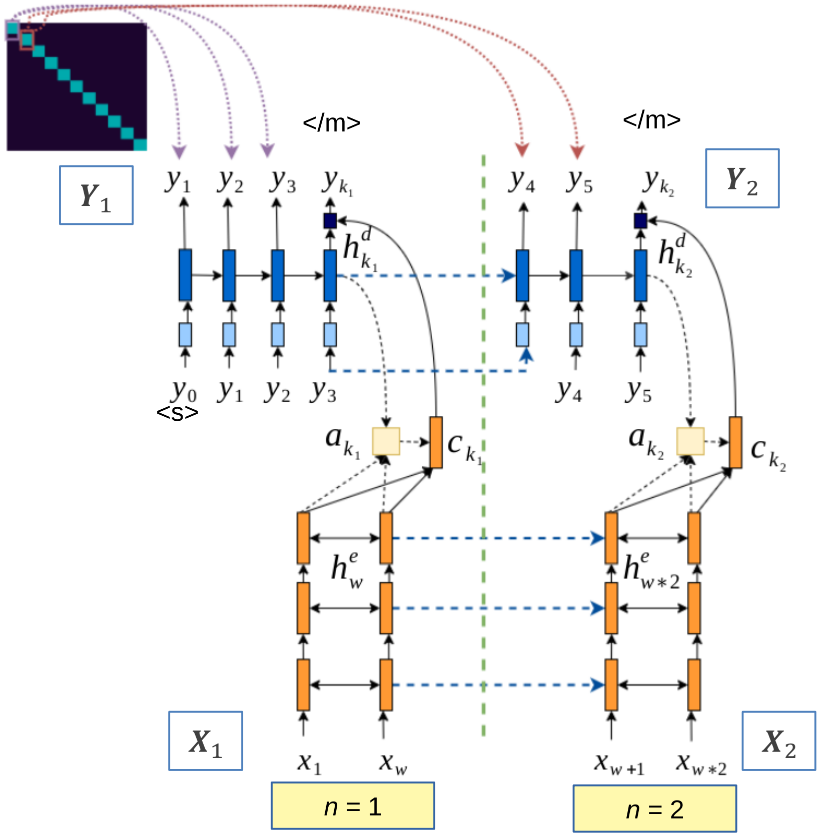


Speech Frame Block ID
(1 block = W frames)

Alignment

Seg. ID (n)	Speech seg. (X_n)	Text seg. (Y_n)
1	$x_1 - x_w$	$y_1 - y_3$
2	$x_{w+1} - x_{w*2}$	$y_4 - y_5$
(etc.)		

2) Train the ISR by using Y_n + </m> as the target of X_n



ISR delay can be managed by changing X_n and Y_n size during training
e.g. higher delay : combine several segments into one

Neural Incremental Speech Recognition

AT-ISR Performance

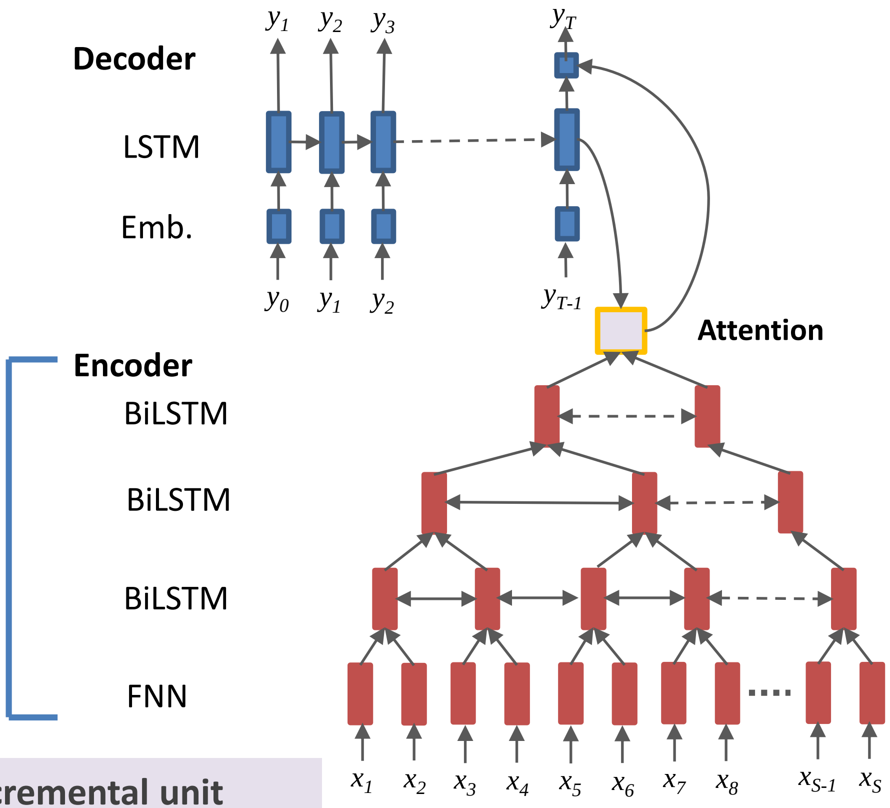
Experiment Dataset

- **Wall Street Journal (WSJ)** [Paul, 1992]
  - 284 speakers, English
  - Training set : *SI-284* set (81 hours of speech)
  - Test set: *eval92* set
- **TED-LIUM release 1** [Rosseau et al., 2012]
  - 118 hours of speech (English)
  - 600 speakers
- Speech features: 80 dim. log-Mel spectrogram (50 ms window, 12.5 ms shift)
- Text token unit
  - Character : Basic Latin alphabet (WSJ, TED-LIUM)
  - Subword : 16,000 subwords (TED-LIUM)

Model Configuration

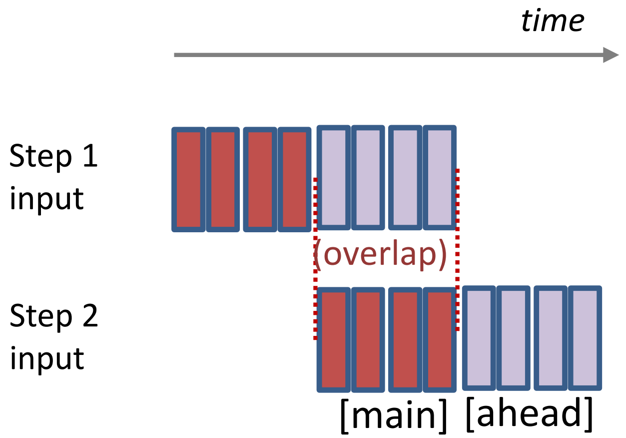
- AT-ISR/Teacher ASR structure: Seq2seq (identical)

Hierarchical sub-sampling
8 speech frames into 1 encoder state



AT-ISR basic incremental unit
8 speech frames = 1 block (0.14 sec)

- AT-ISR with input overlap :
  - Main frames : Aligns with output text seg.
  - Look-ahead frames : Next to the main input (contextual input)



Evaluation Setting

ISR performance evaluation was made by comparing various model:

- **Non-incremental ASR :** Topline
  - Standard seq2seq ASR (Our Att Enc-Dec; teacher)
  - Other existing neural ASR

- **Incremental ASR:**
  - Baseline neural ISR:
    - Seq2seq ISR without attention transfer
    - Incremental steps were taught by using alignments from forced-alignment by HMM ASR
  - Proposed ISR: AT-ISR (attention transfer; student)
  - Other existing neural ISR: Unidirectional LSTM + CTC [Hwang and Sung, 2016]

Evaluation metric:
  - CER, WER
  - Delay (speech input size)

Neural Incremental Speech Recognition
AT-ISR Performance

Speech recognition performance of character-level models trained on WSJ dataset

Model	Delay (sec)		CER (%)
	Input	Computation	
Non-incremental ASR (Topline)			
Att Enc-Dec (ours)	7.88 (avg)	0.32 (avg)	6.26
BiLSTM-CTC [1]			8.97
Joint CTC+Att [1]			7.36
Baseline neural ISR			
Input/step: 1 *m* + 1 *la*	0.24	0.02	20.15
Input/step: 1 *m* + 4 *la*	0.54	0.05	11.95
Proposed AT-ISR			
Input/step: 1 *m* + 1 *la*	0.24	0.02	**18.37**
Input/step: 1 *m* + 4 *la*	0.54	0.05	**7.52**
Other existing neural ISR			
LSTM-CTC beam search [2]	-	-	10.96

CER diff.: 1.3%

Result

- Avg. utterance length: 7.88 sec
- Machine: Intel® CoreTM i7-9700K CPU @ 3.60GHz (NVIDIA GeForce RTX 2080Ti GPU)
- ISR performance limitation: short-segment-based recognition (incomplete information)
- Contextual input (*la*) improves performance

AT-ISR performs well with a short delay by learning non-incremental ASR's knowledge

*Note
m = main input block
la = look-ahead block (contextual input)
1 block = 8 frames = 0.14 sec

[1] Suyoun Kim, Takaaki Hori, and ShinjiWatanabe. Joint CTC-attention based end-to-end speech recognition using multitask learning. In Proceedings of ICASSP, pages 4835-4839, New Orleans, USA, 2017.
[2] Kyuyeon Hwang and Wonyong Sung. Character-level incremental speech recognition with recurrent neural networks. In Proceedings of ICASSP, pages 5335 - 5339, Shanghai, China, 2016.

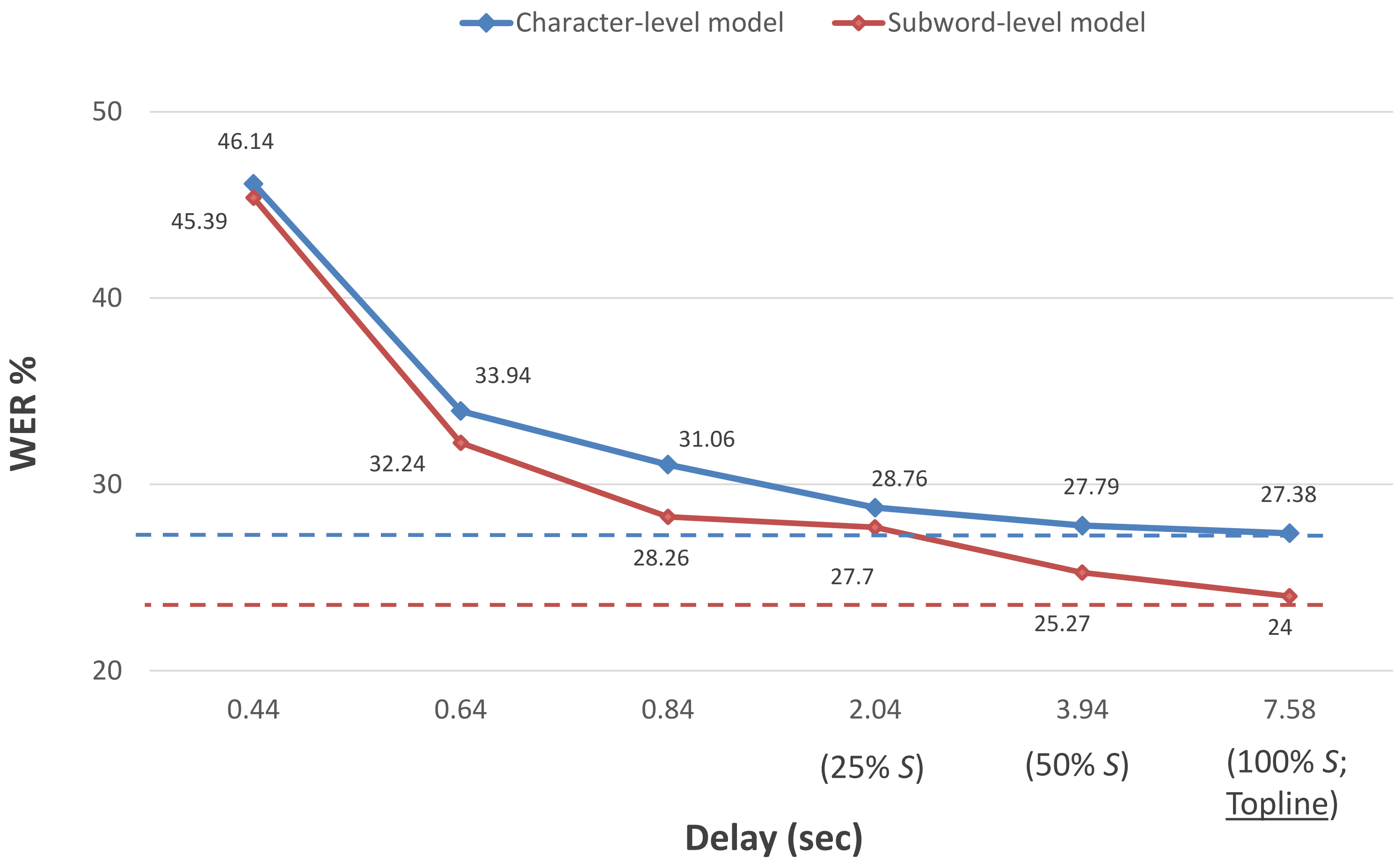
Neural Incremental Speech Recognition
AT-ISR Performance

ISR Delay

How did the ISR delay affected the ISR performance?

- Trade-off: Higher delay, lower WER
- **Subword-level ISR**
  - Lower WER than character-level ISR
  - Keep word context longer than characters
- **Character-level ISR**
  - Maintains the teacher's performance better than the subword-level ISR
  - ISR with delay 2.04 starts to have a close performance to the teacher ASR

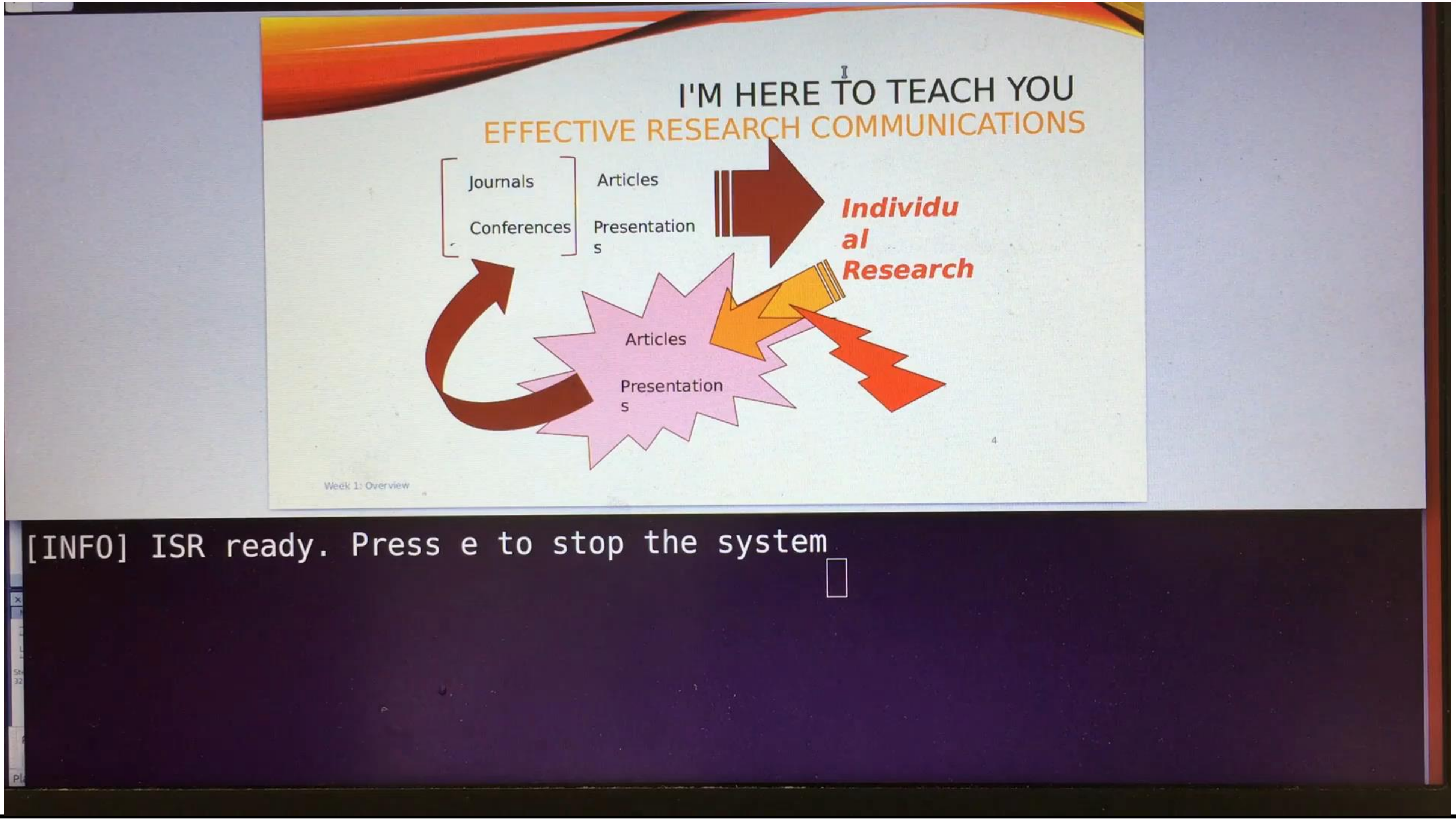
WER (%) of AT-ISR trained on TED-LIUM dataset



*S = average full-utterance length (7.58 sec)

AT-ISR Demo Video – NAIST Lecture

Input segment size/step : 0.84 sec. Machine: Intel ®Core™i7-5500U CPU @ 2.40GHz x 4



Neural Incremental Speech Synthesis

Text-to speech and Incremental Text-to-speech

Text-To-Speech(TTS)

The speech is synthesized Sentence-by-sentence.

1. Input is text or phoneme sequence.
2. Acoustic features are predicted by acoustic model
3. Speech waveforms are reconstructed by vocoder.

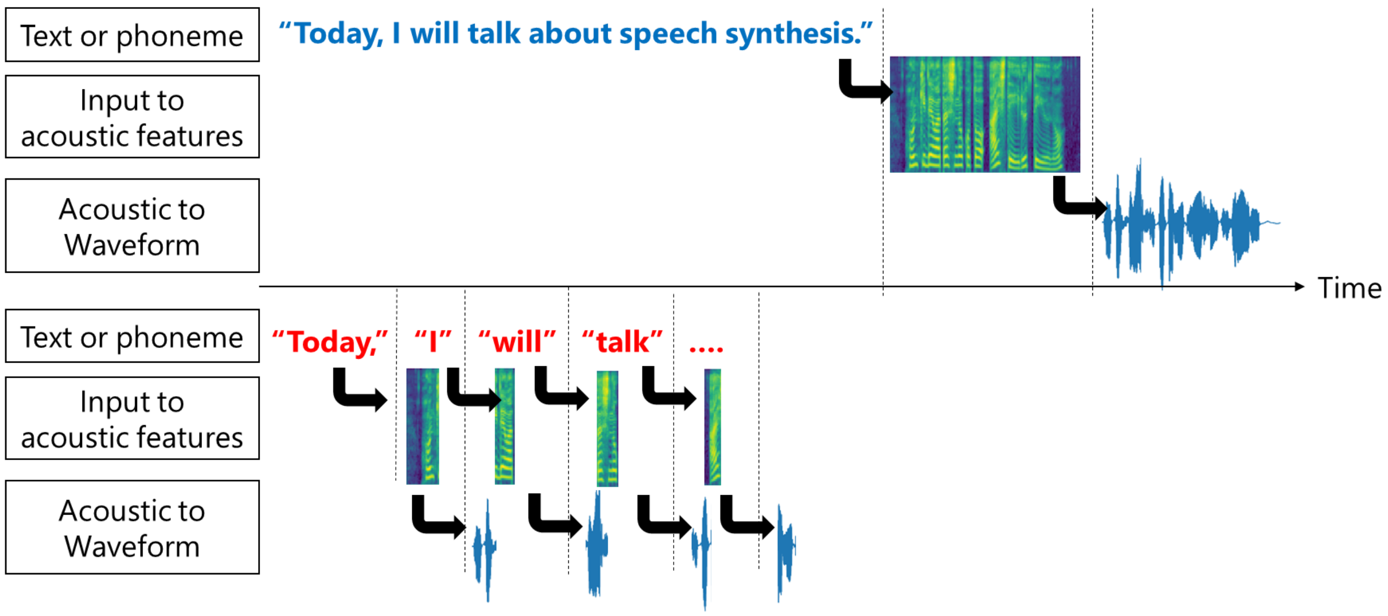
Incremental Text-To-Speech(iTTS)

Speech is synthesizes a speech in shorter delay.

It can synthesize a speech before finishing text input.

Suitable for real-time task

Real-time speech translation

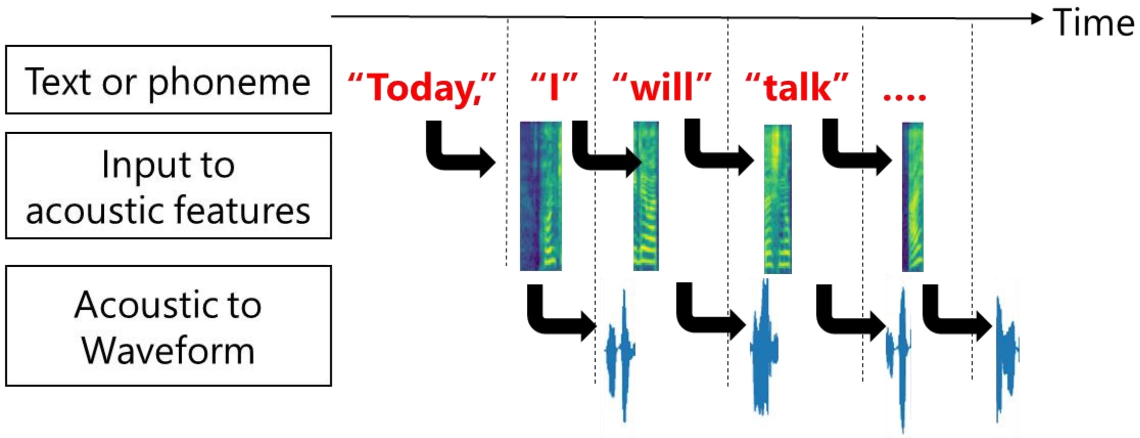


Incremental Text-to-speech

Incremental Text-To-Speech(iTTS)

Speech is synthesizes a speech in shorter delay (e.g. word).

It can synthesize a speech before finishing text input.



Challenges

<u>How to improve speech quality?</u>

Speech quality of Incremental TTS

<u>How to estimate target prosody from an incomplete sentence?</u>

target prosody is typically calculated from long-window features. (e.g. co-articulation)

-> predicts next information(e.g. word) at step of input-to-acoustic-features.

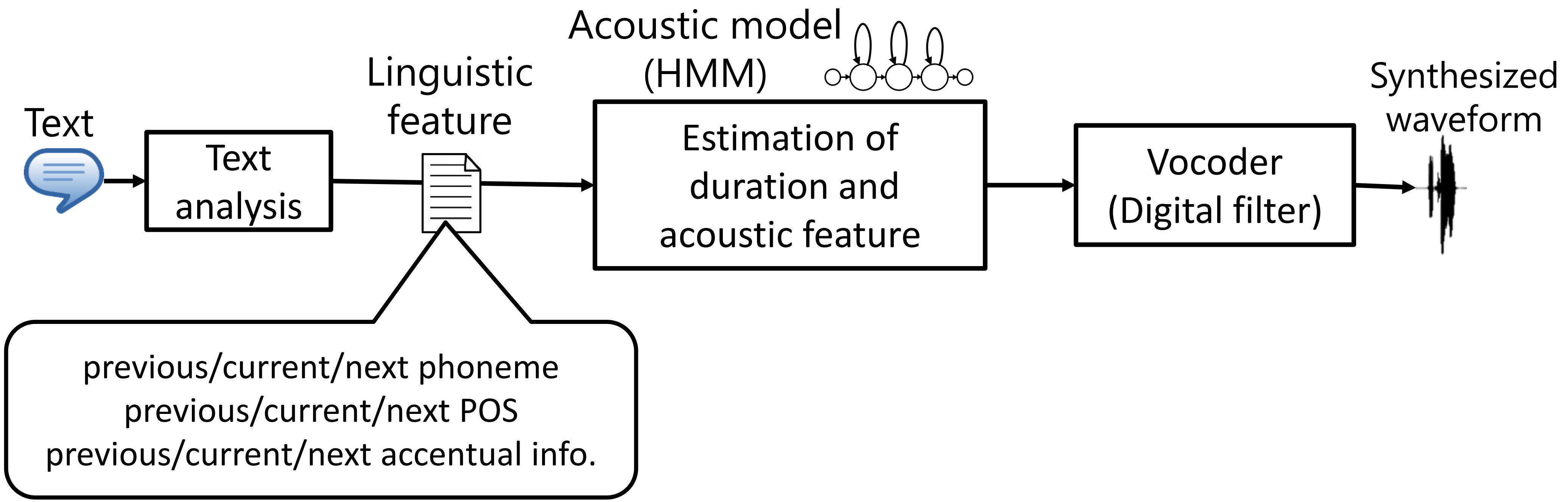
-> wait next word when synthesizing a current word.

Related Works of iTTS(1/2)

Statistical approach (pipeline)

Hidden Markov model TTS[Baumann et al., 2014.],[[Pouget et al., 2015]],[Yanagita, et al., 2018]

No neural End-to-end iTTS approach



Related Works of iTTS(2/2)

End-to-end TTS [Wang, et al., 2017.], [Sotelo, et al., 2017], [Shen, et al., 2018.]

Encoder-decoder with an attention mechanism

-> Output prediction starts after the input sequence.

The speech is also synthesized Sentence-by-sentence.

-> **It can generate High quality speech close to human.**

Challenge of the neural iTTS system.

More natural synthesized speech

Neural iTTS[Yanagita et al., 2019]

no wait next word for synthesis.

Control output sequence with stop flag

Prefix-to-Prefix Framework [Ma et al., 2020]

wait next word for synthesis.

Control output sequence with attention weight and stop flag

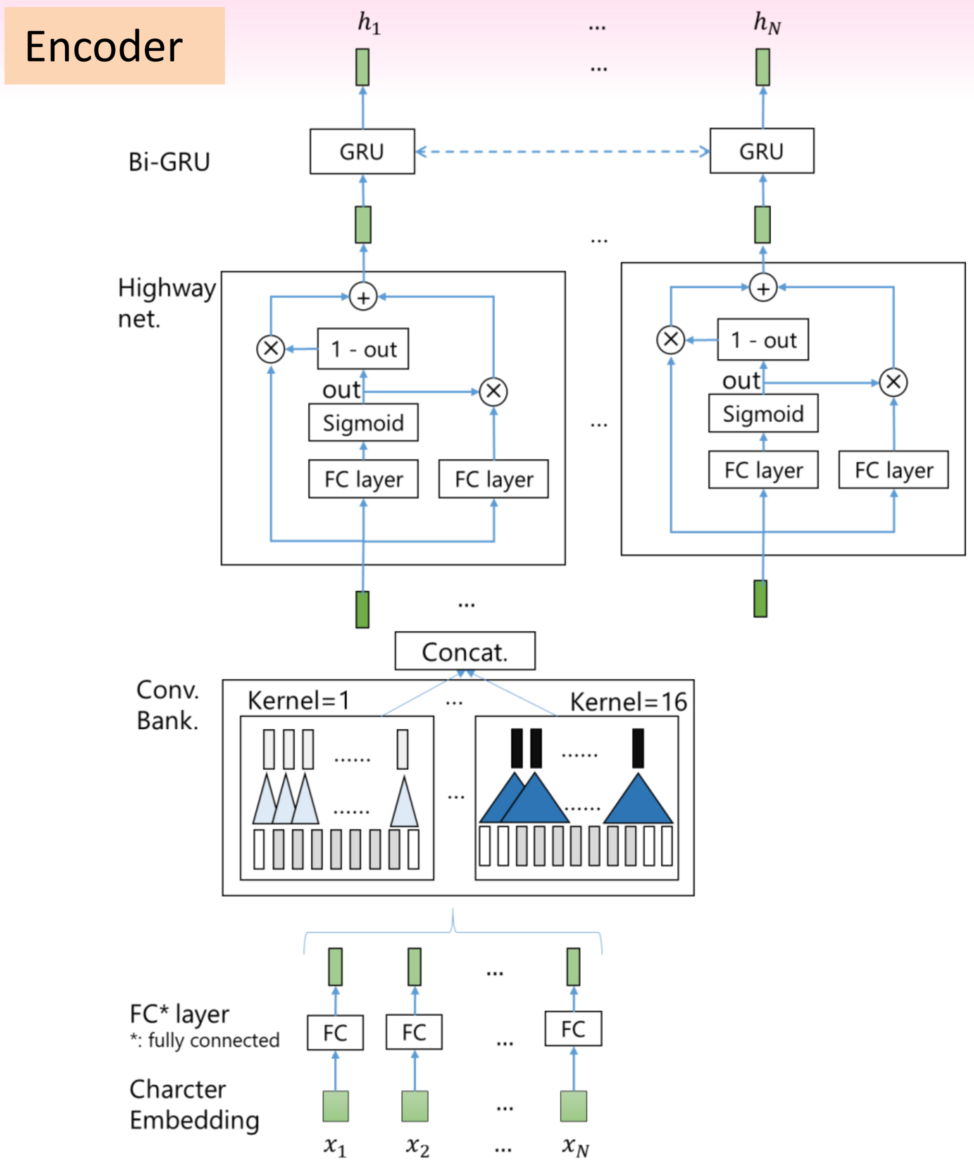
One word look ahead at least for synthesis.

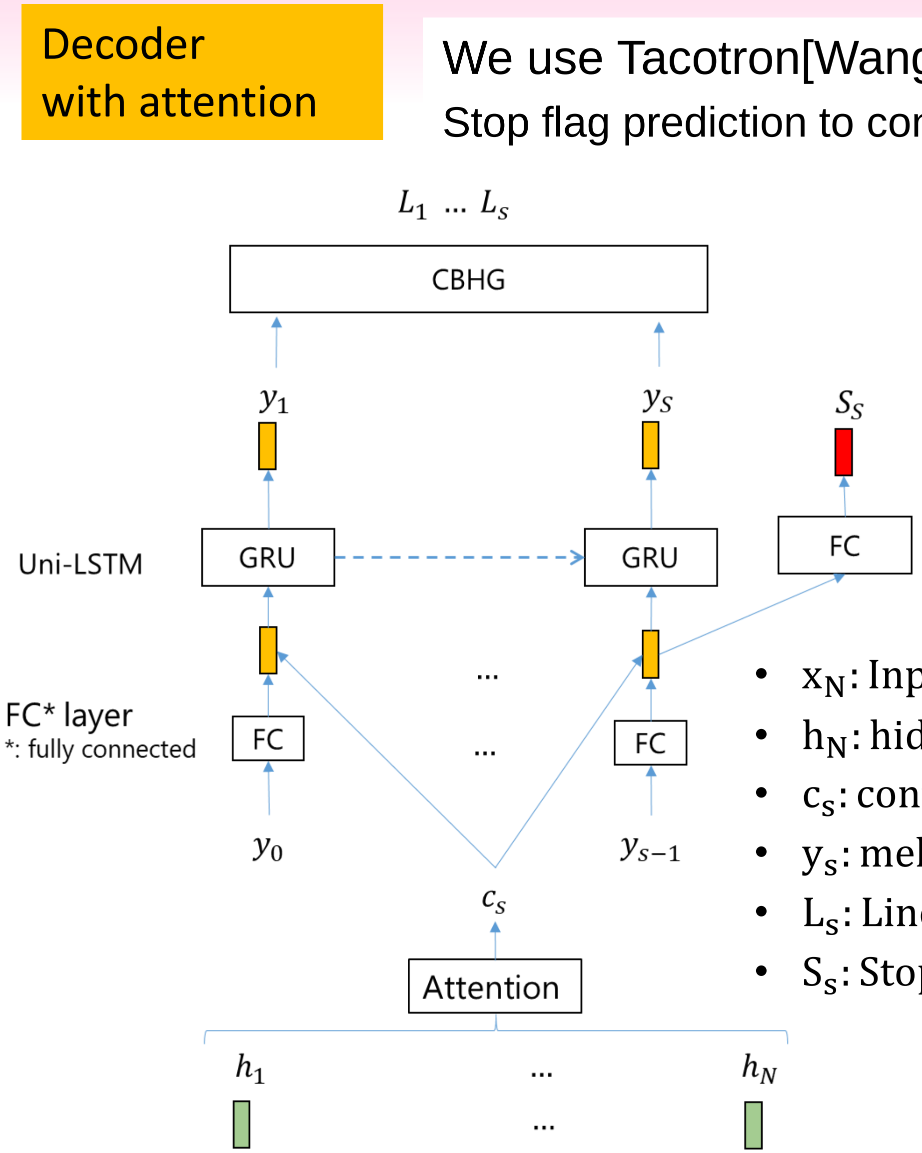
Neural iTTS

Neural iTTS[Yanagita et al., 2019]

Tomoya Yanagita, Sakriani Sakti and Satoshi Nakamura, "Neural iTTS: Toward Synthesizing Speech in Real-time with End-to-end Neural Text-to-Speech Framework", 10th Speech Synthesis Workshop (SSW10) , Sep. 2019

End-to-End TTS





We use Tacotron[Wang, et al., 2017.].

Stop flag prediction to control output seq. is also used.

- x_N: Input sequence (N length)
- h_N: hidden representatiaon of encode
- c_S: context vector (S length)
- y_S: mel spectrogram
- L_S: Linear spectrogram
- S_S: Stop flag

Neural iTTS[Yanagita et al., 2019]

End-to-End iTTS

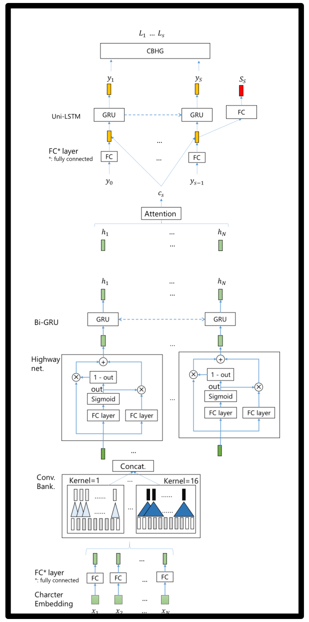
Motivation: We use normal End-to-end TTS as incremental one.

Simple method: Tacotron is synthesized chunk-by-chunk as short sentence.

Ex. “we talk about TTS.”

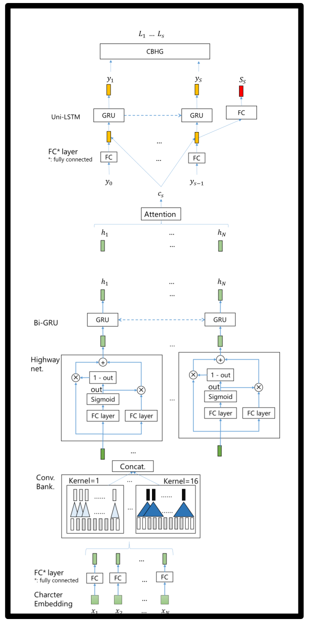
<s>: sentence start, </s>: end of sentence

Tacotron



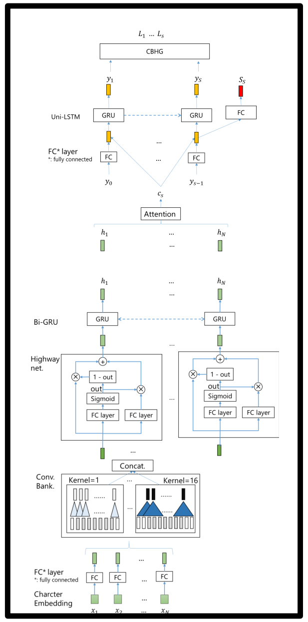
<s> Today </s>

Tacotron



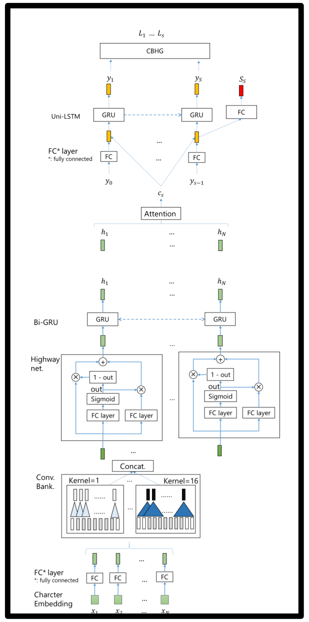
<s> we </s>

Tacotron



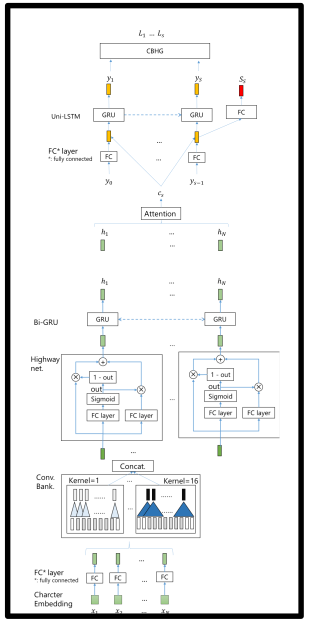
<s> talk </s>

Tacotron



<s> about </s>

Tacotron



<s> TTS. </s>

Proposed Dataset preprocess

<s>: sentence start
</s>: sentence end
<m>: middle sentence start
</m>: middle sentence end

Dataset is divided sentence into three parts.
- use location symbol to indicate locations
- use all data for training

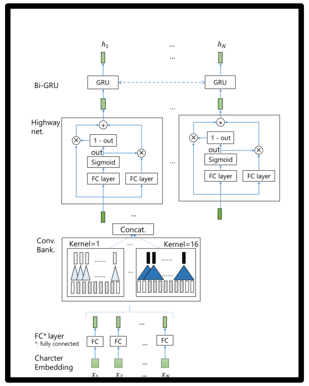
Text and acoustic features

<s>This is Tacotron.</s>
<s>This is an inital step.</s>
<s>Nice to meet you.</s>

<s>This </m> | <m>is </m>
<m>Tacotron.</s>
<s>This is </m> | <m>an </m>
<m>inital step.</s>
<s>Nice </m>
<m>to meet </m> | <m>you.</s>

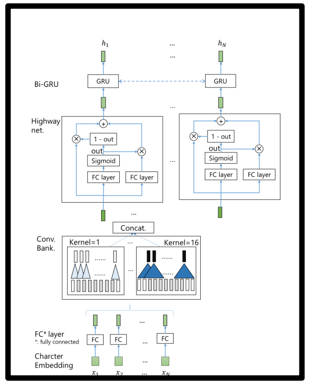
Inference: Ex. “Today we talk about TTS.”

Encoder



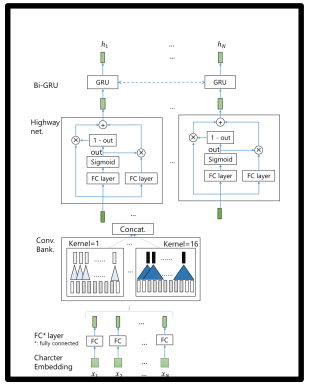
<s> Today </s>

Encoder



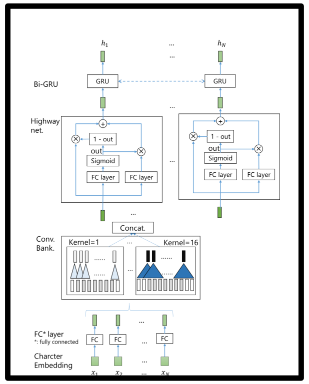
<m> we </s>

Encoder



<s> Today </m>

Encoder



<m> we </m>

Experimental Dataset

- JSUT [https://sites.google.com/site/shinnosuketakamichi/publication/jsut]
  single female speaker, Japanese, 10 hours
  Training set: 5k utterance
  Test and dev: 100k utterance

- LJ-speech [https://keithito.com/LJ-Speech-Dataset/]
  single female speaker, English, 24 hours
  Training set: 10k utterance
  Test and dev: 100k utterance

- Input sequence
  Phoneme and accentual information (Japanese)
  Word character (English)

- Preprocess dataset
  Ja. Dataset is divided sentence into three parts in the basis of phrase position.
  En. Dataset is divided sentence into three parts in the basis of word position.

- Acoustic features: 80 dim. mel-spectram, 1024 dim. Linear-spectrogram

Evaluation method

We concatenated all the synthesized waveforms into sentence-based waveforms.

- Synthesis various input length (e.g. word-by-word, 2words-by-2words)
- To compare to normal TTS waveforms

Evaluation methods

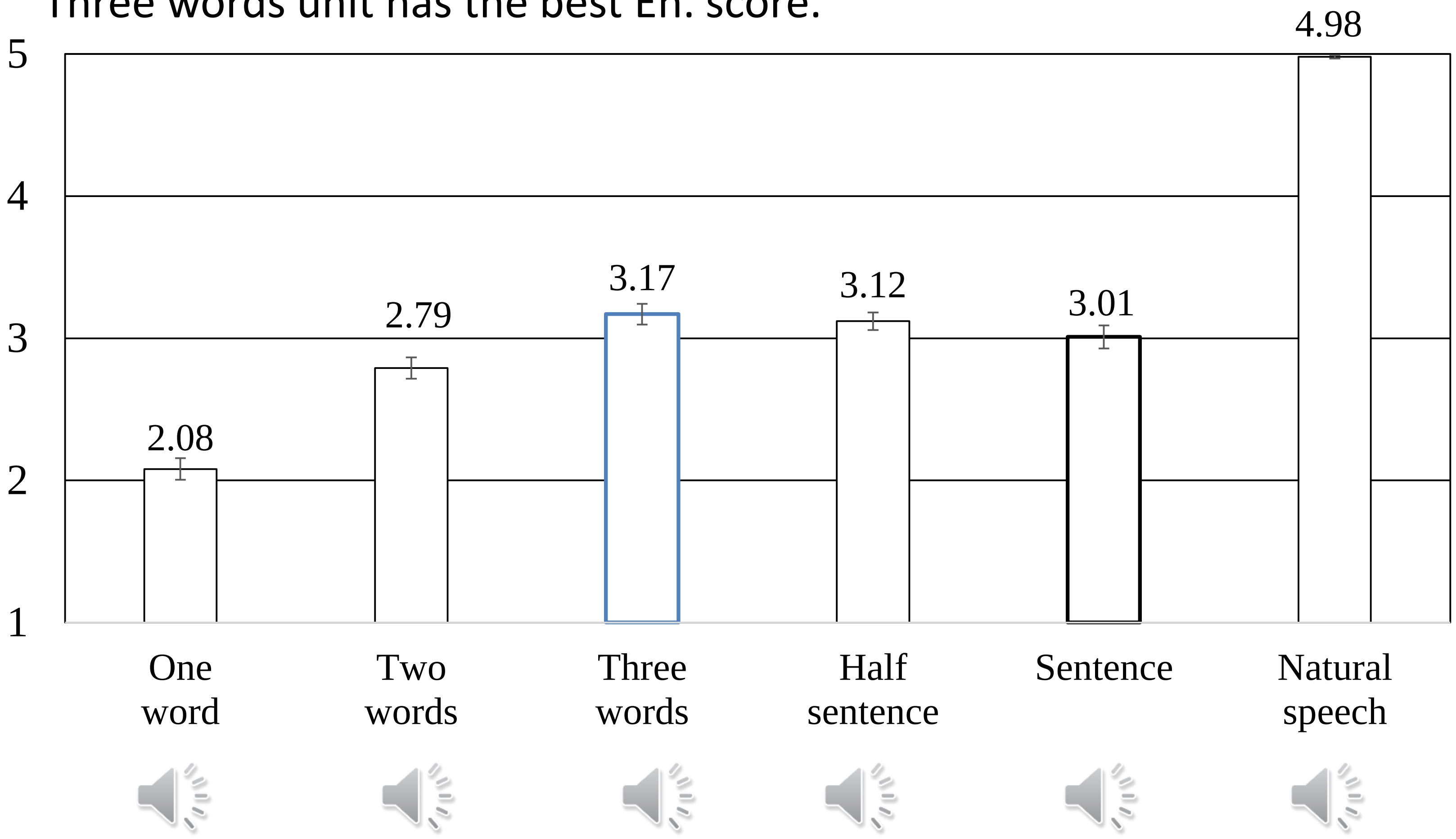
MOS test for naturalness

- Evaluator listens one waveform
  and score 5 scales(1: very bad, 2:bad, 3:normal, 4: good, 5: very good)

Result of English MOS

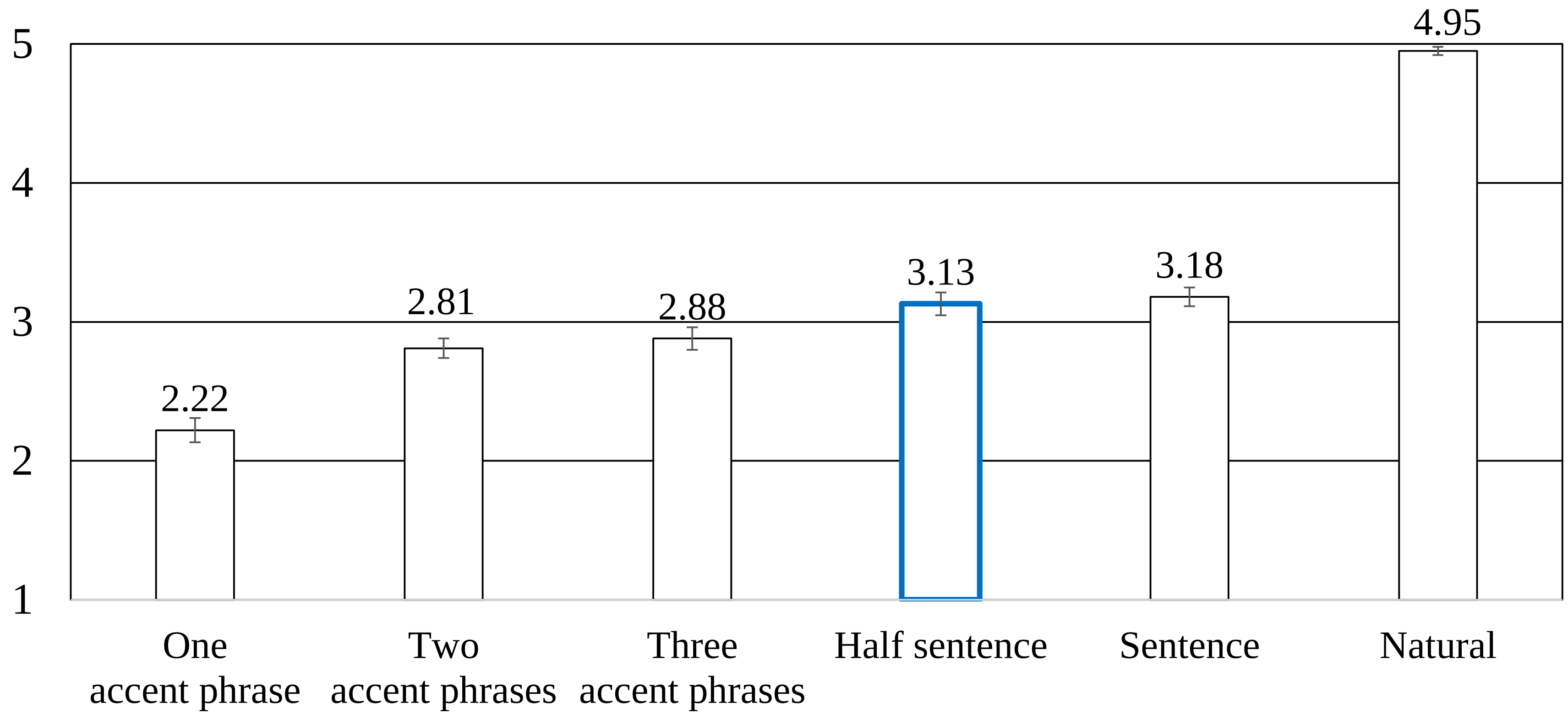
Still big gap between natural speech and synthesized speech.

Three words unit has the best En. score.



Result of Japanese MOS

Still big gap between natural speech and synthesized speech.
Half sentence unit ≒ the full sentence units (Ja.).



Neural Incremental Speech Recognition

Summary – AT-ISR

Neural ISR system (AT-ISR) with a low recognition delay without increasing the complexity of the standard ASR system

1. AT-ISR with delay < 1 sec. achieved a close performance to standard ASR with delay > 7 sec.
2. AT-ISR as an ISR framework with an efficient development mechanism and reliable performance via attention transfer that applies an identical architecture as the standard ASR

Recent ISR Trend

- Streaming ASR with RNN-Transducer (RNN-T) [Saitnah et al., 2020; Li et al., 2020]
- Streaming transformer ASR [Miao et al., 2020; Moritz et al., 2020; Tsunoo et al., 2020]

iTTS Summary

Incremental End-to-end TTS

Previous work: HMM-based iTTS

We challenge neural iTTS system by extending conventional neural TTS

-> add location symbols for input

-> use initial input for decoder

Future work

The wide gap between natural speech and synthesized speech.

-> wavenet vocoder

Improvement of stop flag prediction for English model

-> very short sentence (e.g. “It”)

Calculation of delay

NAIST AHC Lab.