Sequence-to-Sequence Learning via Attention Transfer for Incremental Speech Recognition

Sashi Novitasari[1], Andros Tjandra[1], Sakriani Sakti[1,2], Satoshi Nakamura[1,2]

[1] Nara Institute of Science and Technology (NAIST), Japan
[2] RIKEN Center for Advanced Intelligence Project (RIKEN AIP), Japan
{sashi.novitasari.si3, andros.tjandra.ai6, ssakti, s-nakamura}@is.naist.jp

Outline

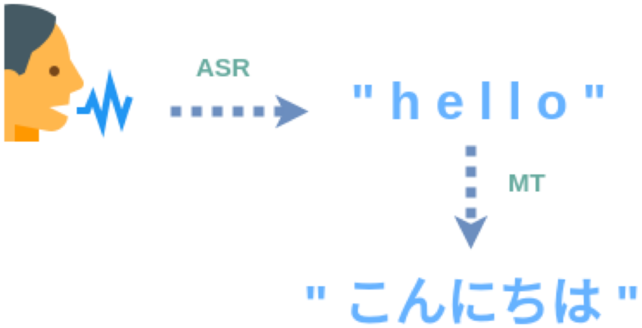
I. Background

Simultaneous Speech Translation

- Translate incoming speech to the target language in real-time with low delay (incremental)

- Examples of use
  - Meeting
  - Lecture talk
  - Live video

- **Automatic** → require ASR that can recognize the speech immediately after the original timing



Automatic Speech Recognition

Generate transcription of a speech utterance

- **Non-incremental ASR**
  - Wait for the speech to finish first
  - *State-of-the-art* ASR: Att Enc-Dec (end-to-end)

  → Not suitable for simultaneous translation

- **Incremental ASR (ISR)**
  - Recognize **without** the need for **waiting** for the **speech to finish** [Selfridge et al., 2011]
  - Part-by-part recognition

  → Suitable for simultaneous translation

Incremental Speech Recognition

- HMM-based ASR is incremental but not end-to-end
- Seq2seq ISR : train by learning input-output parts alignments (e.g. Neural transducer [Jaitly et al., 2016])
- **End-to-end seq2seq ISR** → more **complex** training than standard seq2seq ASR
  - Learn the incremental step?
  - Ground alignments?
    - Generate it during training based on ISR model (multiple times)
    - Generate it by using external module (once)

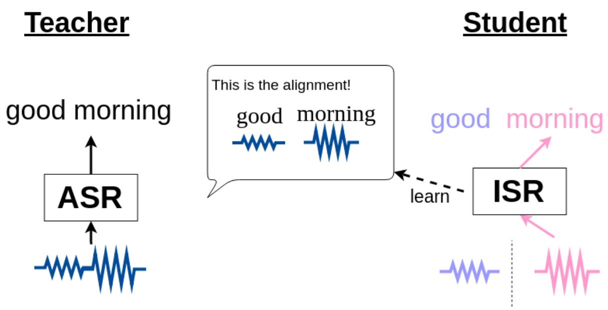
→ <u>Expensive</u>
(especially if module not available)

How to make reliable ISR with simple method?

Goal

Attention Transfer Incremental Speech Recognition (AT-ISR)

- Simple training & recognition → **Exploit attention-based seq2seq ASR**
  - ISR architecture : Att Enc-Dec ASR (seq2seq)
  - Incremental step & alignment : Learn the attention knowledge from ASR
    → **attention transfer**
- **Attention transfer** : Attention knowledge transfer from teacher to student
  - 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



AT-ISR
ISR that learns to mimic attention-based alignment from attention-based ASR

II. AT-ISR

AT-ISR Recognition Method

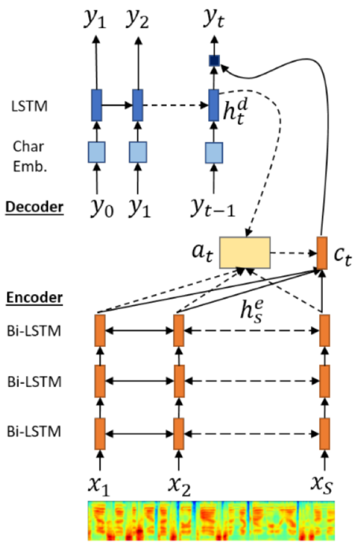
- Att Enc-Dec
- Input segment → *W* frames, consist of:
  - *M* frames → main input
  - *C* frames → contextual input (optional, adjacent to main input)
- Recognize segment-by-segment sequentially
  - For **each recognition step**:
    1. **Encode** *W* speech frames (block)
    2. **Decode** for the output that aligned to the main input block, until *end-of-block* token predicted or max. length reached
       2.1 **Attend** the current input
    3. Shift the input window *M* frames
- How to learn the *end-of-block* (*</m>*)? → Attention transfer

Encoder-Decoder with Attention Mechanism

3 main parts:

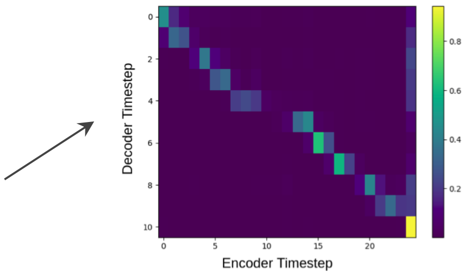
- **Encoder**
  Encode input features
- **Decoder**
  Decode encoded information into output
- **Attention**
  Calculate alignment score between encoder states (input) and decoder states (output)
  → attention matrix

Att Enc-Dec



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

Attention Matrix



Learning the Alignment

Attention Transfer for ISR

Train ISR (student) to learn the attention-based alignment from attention-based ASR (teacher)

Attention-based alignment

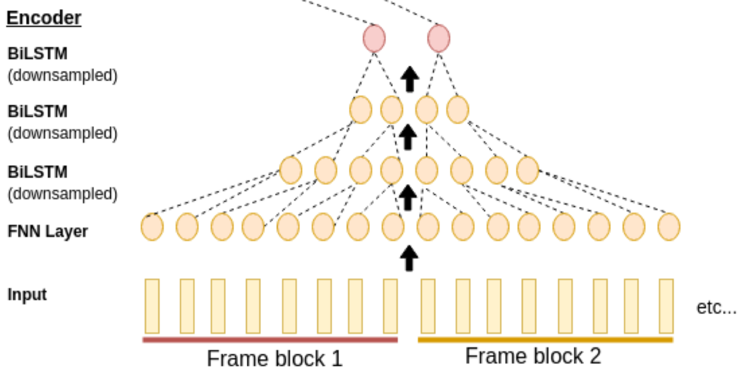
- 1 encoder state represents *M* input frames
  → 1 output aligned to *M* frames
- *M*: downsampling rate in encoder

Ground Alignment for ISR training

- Token aligned to an encoder state with highest attention alignment score (monotonic)
- *</m>* placed after each last-aligned token in a segment
- Alignment generation by teacher-forcing

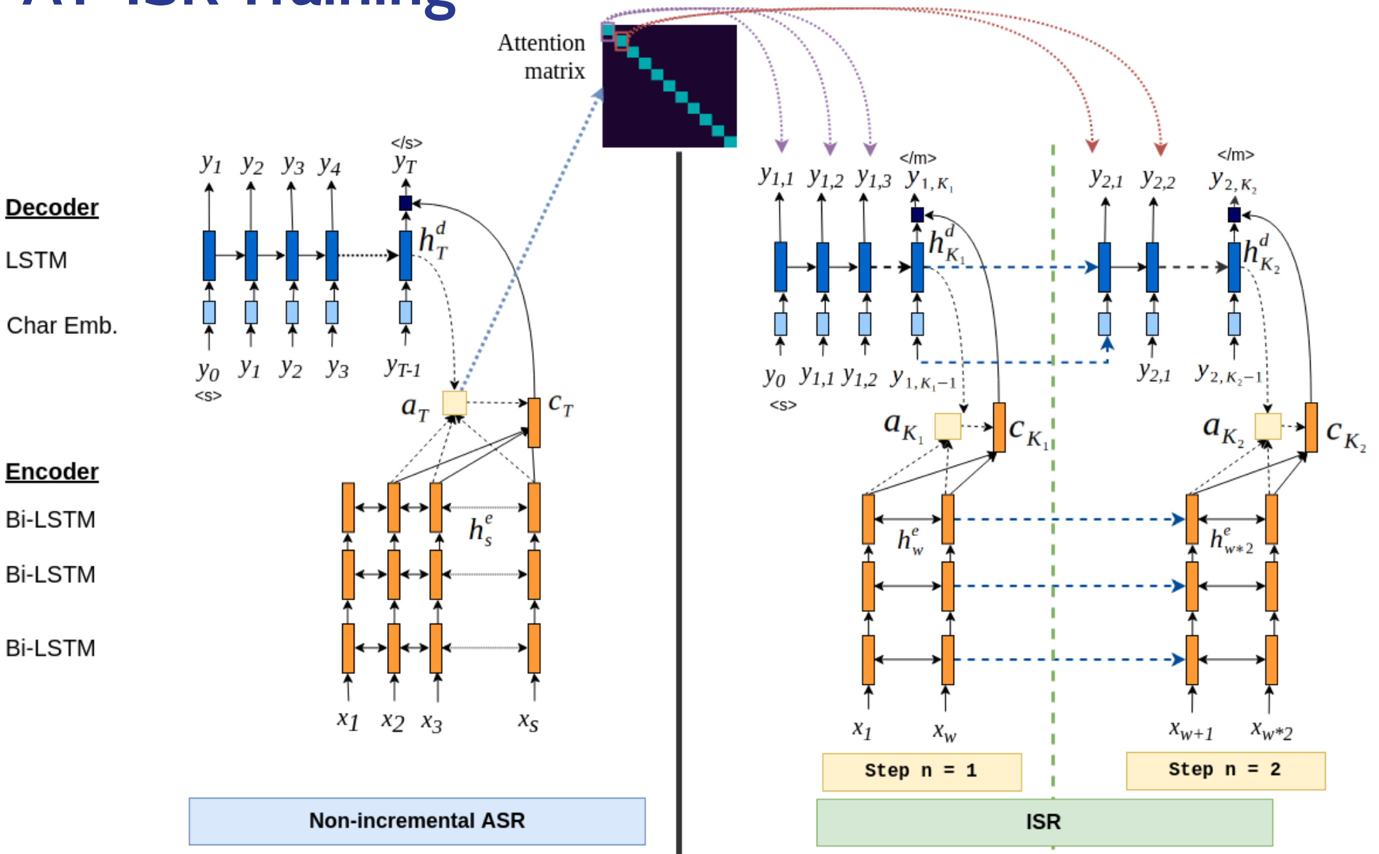
Attention Matrix

	enc_state1	enc_state2	enc_state3	etc..
char1	0.94312793	0.020486057	0.011381225	[illegible]
char2	0.663976312	0.117226064	0.062948689	0.015263585
char3	0.41076684	0.213412568	0.091481216	0.033991031



char1, *char2*, *char3* aligned to *enc_state1* = *frame block 1* (*M* = 8 frames)

AT-ISR Training



Given

Speech frames **X** = [$x_1, x_2, \ldots x_S$]

Transcription **Y** = [$y_1, y_2, \ldots, y_T$]

Non-incremental ASR

P(**Y** | **X**)

AT-ISR

For each step *n*:

P(**Y**$_n$ | **X**$_n$)

where:

- **X**$_n$ = [$x_{((n-1)w)+1}, \ldots, x_{nw}$]
- **Y**$_n$ = [$y_{n,1}, \ldots, y_{n,Kn}$]
- $y_{n,Kn}$ = </m> token
- $0 <= K_n <= K < T$
- **Y**$_n$ aligned to **X**$_n$ (attention alignment)

III. Experiments

Data and Features

- Dataset (English)

Dataset	Speaker	Length (hour)	Expr. Sets (utterance)		
			Train	Dev.	Test
LJ Speech [Ito et al., 2017]	1	24	12000	400	400
Wall Street Journal [Paul et al., 1992]	80 (*si84*) 280 (*si284*)	16 (*si84*) 80 (*si284*)	7000 (*si84*) 37000 (*si284*)	500 (*dev93*)	300 (*eval92*)

- Features
  - 80-mel spectogram
  - Window length 50 ms, shift 12.5 ms
- Output representation: Character (basic Latin alphabet)

Model Configuration

Non-incremental ASR & AT-ISR → **Att Enc-Dec** (parameters based on [Tjandra et al., 2017]):

- **Encoder**
  - 1 FNN layer (256 units) , 3 BiLSTM layers (256 units/LSTM layer)
  - Downsampling: 2 states for each BiLSTM layer
    - Final encoder states represent 8 frames each
    - ISR input block unit: 1 block = 8 frames = ~0.14 sec

- **Decoder** : 1 embedding layer (256 units) , 1 LSTM layer (512 units)

- **Attention :** MLP-scoring with multi-scale alignment and contextual history [Tjandra et al., 2018]

- No language model

Experiment Scenario

Topline : Non-incremental recognition by teacher ASR
Baseline : Incremental recognition by teacher ASR (no attention transfer)

Experiments:

1) Mechanism configuration
   How to take encoder and decoder input, how to treat model states

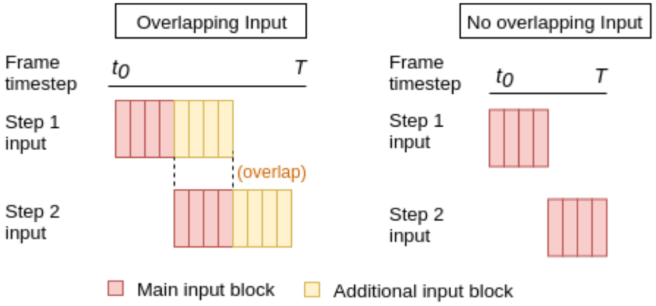
2) Delay
   How the AT-ISR delay affects the performance

Experiment 1
Mechanism Configuration

Encoder Input

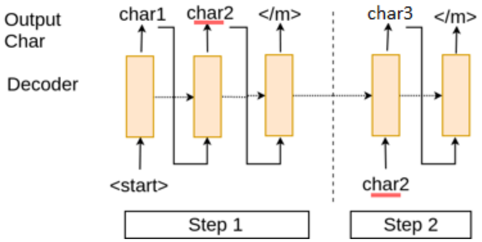
a. **Overlapping inputs**
- Include <u>context</u> blocks in addition to the main block (adjacent):
  - Look-back : prev. to the main block
  - Look-ahead : next to the main block
- Output → tokens that aligned to the main block

b. **No overlapping input**

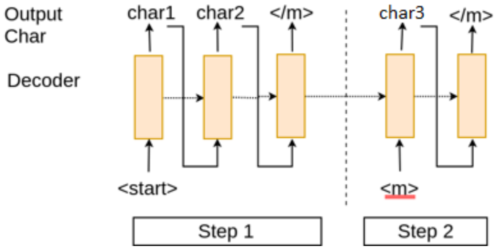


Decoder Initial Input

a. **Last token** from the prev. step (before *</m>*)



b. **Beginning-of-block** *<m>*



Experiment 1
Result

- AT-ISR model states:
  - **Reset** at the beginning of step
  - **Keep** the states from previous step

- ISR input segment size in each step:
  - 1 main block (~0.14 sec)
  - Overlap: +1 look-ahead block

- **Best mechanism:**
  - Encoding : Input overlap
  - Decoding : Last character from prev. step as initial input
  - Model states: Keep

→ as default mechanism for AT-ISR

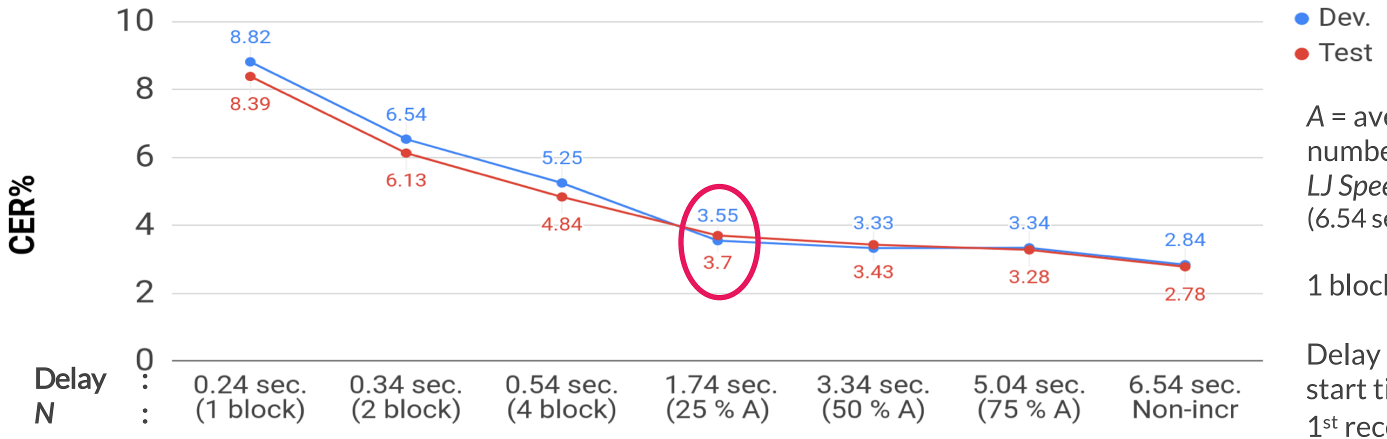
Utterance-based CER% on LJ Speech Dataset

Enc-Inp	Dec. Initial Inp	Delay (sec)	Dev.	Test
Topline ASR		6.54 (avg.)	2.84	2.78
Baseline ISR		0.14	79.63	80.34
AT-ISR - reset state				
No overlap	*<m>*	0.14	32.51	32.35
No overlap	prev. char	0.14	26.15	26.52
Overlap	*<m>*	0.24	23.74	23.40
Overlap	prev. char	0.24	13.40	14.22
AT-ISR - keep state				
No overlap	*<m>*	0.14	24.35	24.44
No overlap	prev. char	0.14	22.69	23.04
Overlap	*<m>*	0.24	8.83	8.16
Overlap	prev. char	0.24	8.82	8.39

Experiment 2
Delay: Main block

- LJ Speech dataset
- Tradeoff : Higher delay → higher performance
- Insignificant improvement after certain delay conf. → **shortest delay with best performance**

Impact of Main Input Block Size on Utterance-based CER
AT-ISR Input: [*N* main + 1 ahead] blocks/step



A = average frame block number in an utterance in *LJ Speech* set (6.54 sec~ 520 blocks)

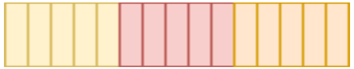
1 block = 8 frames = ~0.14 sec

Delay = Time diff. of speech's start timing and start timing of 1st recognition step

Experiment 2

Delay: Context blocks

AT-ISR Input segment

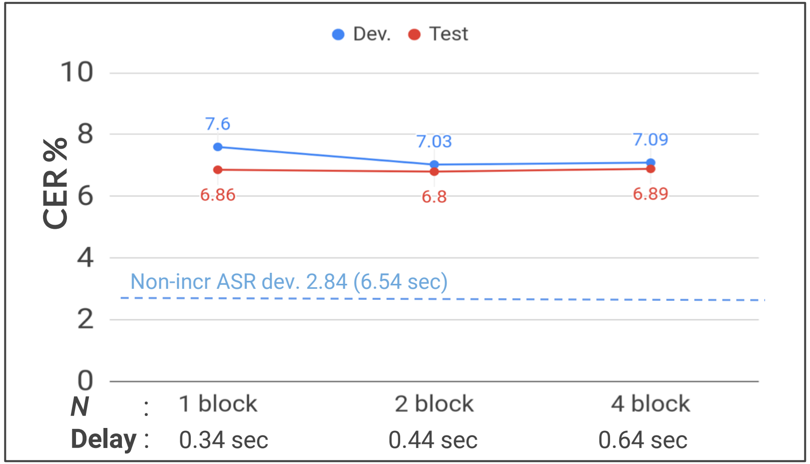


L.back block (context) | Main block | L.ahead block (context)

- LJ Speech, utterance-based CER
- Without context block → CER 22.7% (dev.)
- Context blocks **help** the recognition, especially **look-ahead** blocks

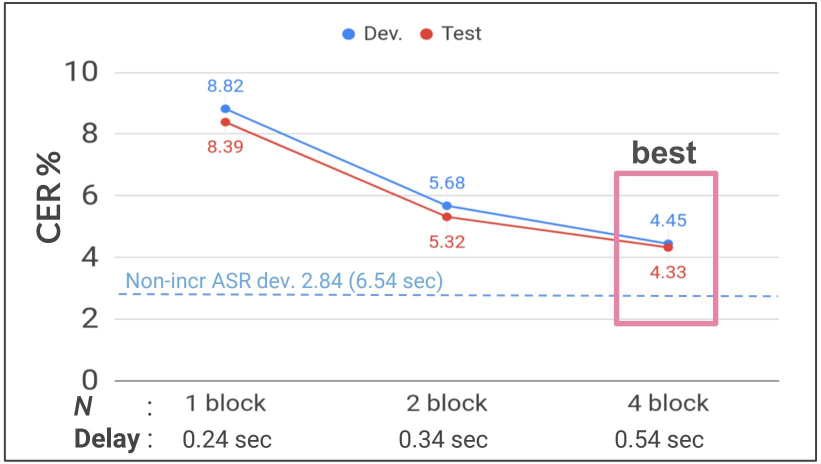
Impact of look-back block

AT-ISR input: [*N* back + 1 main + 1 ahead] blocks/step



Impact of look-ahead block

AT-ISR input: [1 main + *N* ahead] blocks/step



*1 block = 8 frames = ~0.14 sec

Performance on Multi-speaker Data

Utterance-based CER (%) on *eval92* Set

Non-incremental ASR (Topline)				
Model		Delay	*si84*	*si284*
CTC [Kim et al., 2017]		7.5 sec (avg.)	20.34	8.97
Att Enc-Dec Content [Kim et al., 2017]			20.06	11.08
Att Enc-Dec Location [Kim et al., 2017]			17.01	8.17
Join CTC+Att (MTL) [Kim et al., 2017]			**14.53**	7.36
Att Enc-Dec (Teacher)			17.05	**6.80**
AT-ISR (1 main input block/step)				
Look-back/step	Look-ahead/step	Delay	*si84*	*si284*
0 block	1 block	0.24 sec	30.81	19.78
0 block	4 block	0.54 sec	**18.05**	**9.06**

*1 block = 8 frames = ~0.14 sec

- WSJ dataset
- Train set:
  - *si84* : 80 speakers
  - *si284* : 280 speakers
- WSJ *si284* model→ delay 0.54 sec, CER difference to teacher ~2%
- AT-ISR able to perform closely to teacher on multi-speaker data

IV. Conclusion

Conclusion

- Incremental speech recognition with AT-ISR -- ISR that learned the same attention alignment as the teacher non-incremental ASR

- AT-ISR able to perform closely to the teacher by incrementally recognizing short input segments (low latency and reliable)
    - LJ Speech CER → teacher 2.84% ; student 4.45% (delay 0.54 sec)
    - WSJ CER → teacher 6.80% ; student 9.06% (delay 0.54 sec)

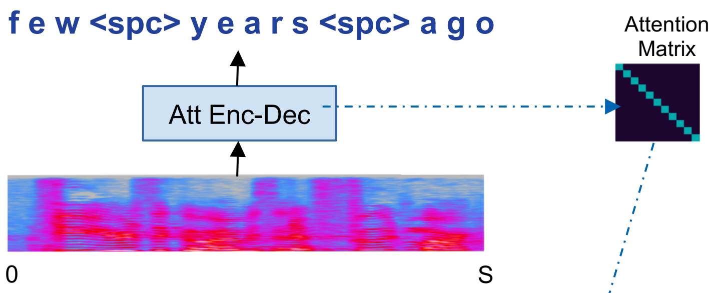
- Optimum AT-ISR performance achieved by, for each step, including few ahead blocks and setting the last character from the last step as the initial input in decoding

Thank You

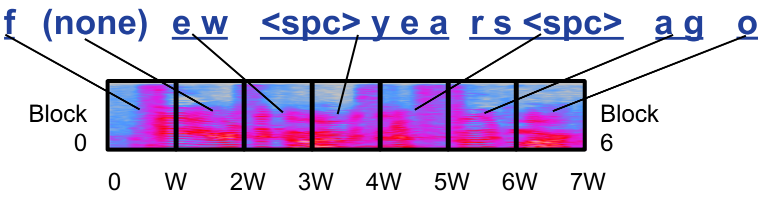
Example

Non-incremental ASR

Text generation:



Alignment based on attention scores:

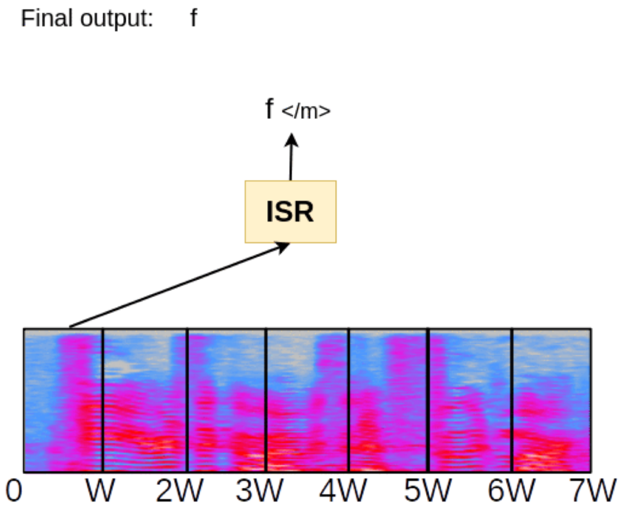


Example of alignment index calculation for the 6th character *'y'*

(*k* → 1 enc state = 1 frame block = *W* frames (downsample))

AT-ISR

Training: Encode *W* frames, decode aligned chars+*</m>* as the target



Text generation: Encode *W* frames and decode until *</m>*, *</s>*, or reach max. length

TED-LIUM

- Unk. word : Rate of words that does not exists in the train data (eval. set original text = 1.55%)
- Character-to-subword:
  - Sentencepiece:
    - Wait for 1 word then convert it into subwords (1 word = 8 characters (avg.))
    - Same CER, WER, and unk. word rate as the character-level ISR
  - Seq2seq:
    - Incremental: convert 8 characters by looking 8 characters ahead
    - Speech-character and character-subword models trained separately

Performances (%) on TED-LIUM release 1

ISR input-output	CER	SWER	WER	Unk. word
Full-utterance ASR (avg. delay: 7.58 sec)				
sp-ch-sw (sentencepiece)	15.21	20.16	27.37	3.02
sp-ch-sw (seq2seq)	15.81	20.46	28.32	1.03
sp-sw	**13.35**	**18.91**	**23.98**	**0.62**
ISR (input segment: 1 main + 4 ahead bocks →delay: 0.54 sec)				
sp-ch-sw (sentencepiece)	**21.00**	31.87	41.10	11.7
sp-ch-sw (seq2seq)	22.36	27.53	39.71	1.34
sp-sw	21.28	**25.70**	**36.78**	**0.66**
ISR (input segment: 4 main + 4 ahead bocks → delay: 0.84 sec)				
sp-ch-sw (sentencepiece)	16.22	23.11	31.04	5.19
sp-ch-sw (seq2seq)	17.99	22.60	31.80	1.66
sp-sw	**15.20**	**19.88**	**28.26**	**1.04**

sp : speech features
ch : character
sw : subword
sp-ch-sw: char-level ISR and character-subword model