Towards Streaming Speech Translation

Javier Iranzo-Sánchez

jairsan@vrain.upv.es

www.mllp.upv.es

Joint work with MLLP researchers

MLLP | Machine Learning and Language Processing

VRAIN

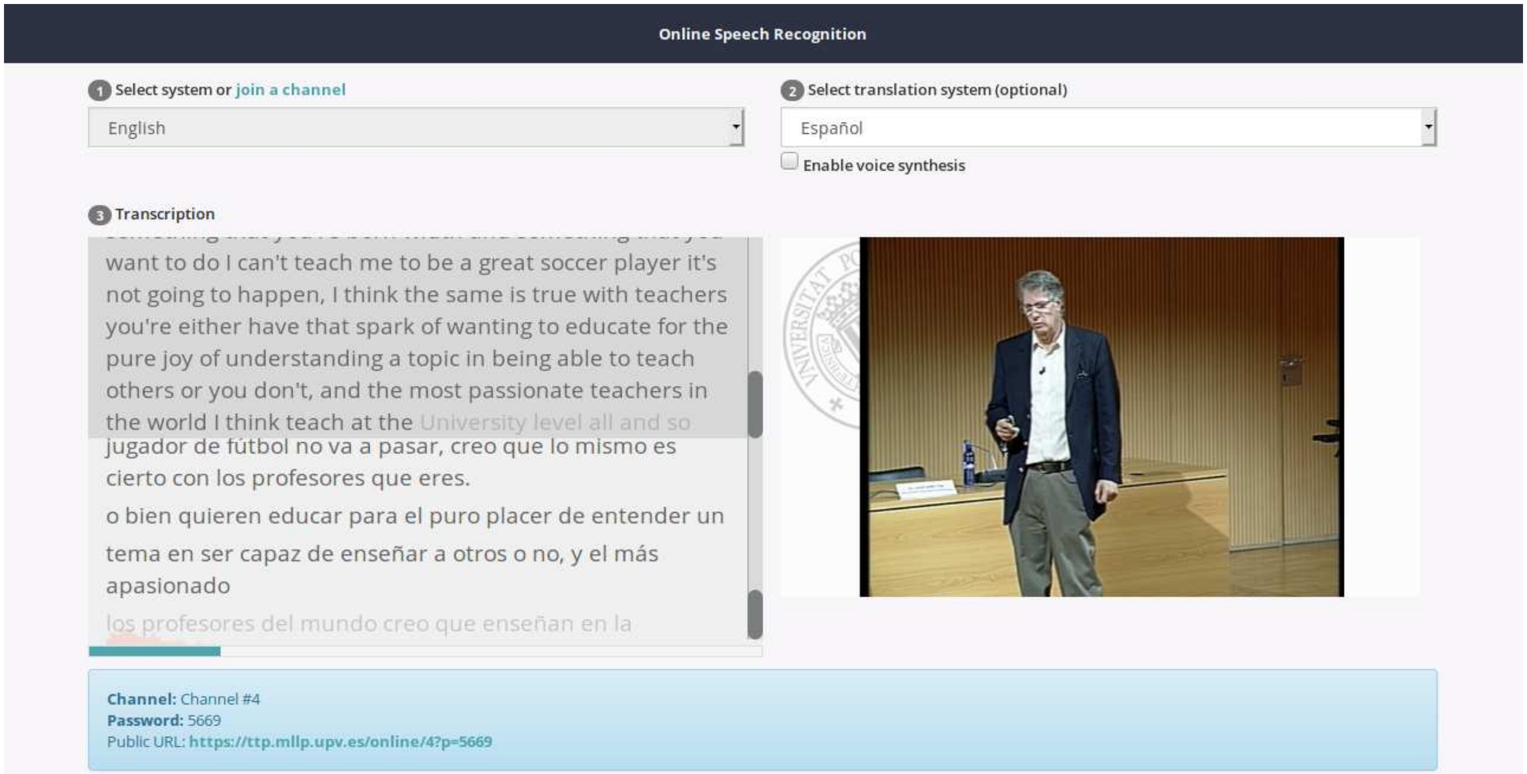
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Contents

1 Introduction

Streaming Speech Translation

- ASR + MT



  - ▷ Translate an unbounded input audio stream
  - ▷ Real-time factor constraints
  - ▷ Latency constraints → Partial input
  - ▷ Stream-level evaluation

2 Streaming ASR

Streaming ASR poses several challenges:

- Processing and providing output in ***real-time***
- ***Limited context*** to perform the recognition

Goals:

- Language model: Speed-up computations
- Acoustic model: Limited context to compute features

Proposed techniques:

- Language model: One-pass decoder based on LSTM LM
- Acoustic model: BLSTM with a future context window

One-pass decoder review

Real-time One-pass Decoder for Speech Recognition Using LSTM LM [Jorge et al., 2019]

Challenges:

- One-pass decoder
- RFT $\leq$ 1.0

Proposed techniques:

- Decoder structure based on LM histories
- LSTM LM *on-the-fly* rescoring
- Softmax complexity's reduction w/ appr. denominator (VR)
- New parameters to control the WER/RTF trade-off

Streaming one-pass decoder

LSTM-Based One-Pass Decoder for Low-Latency Streaming [Jorge et al., 2020]

Challenges:

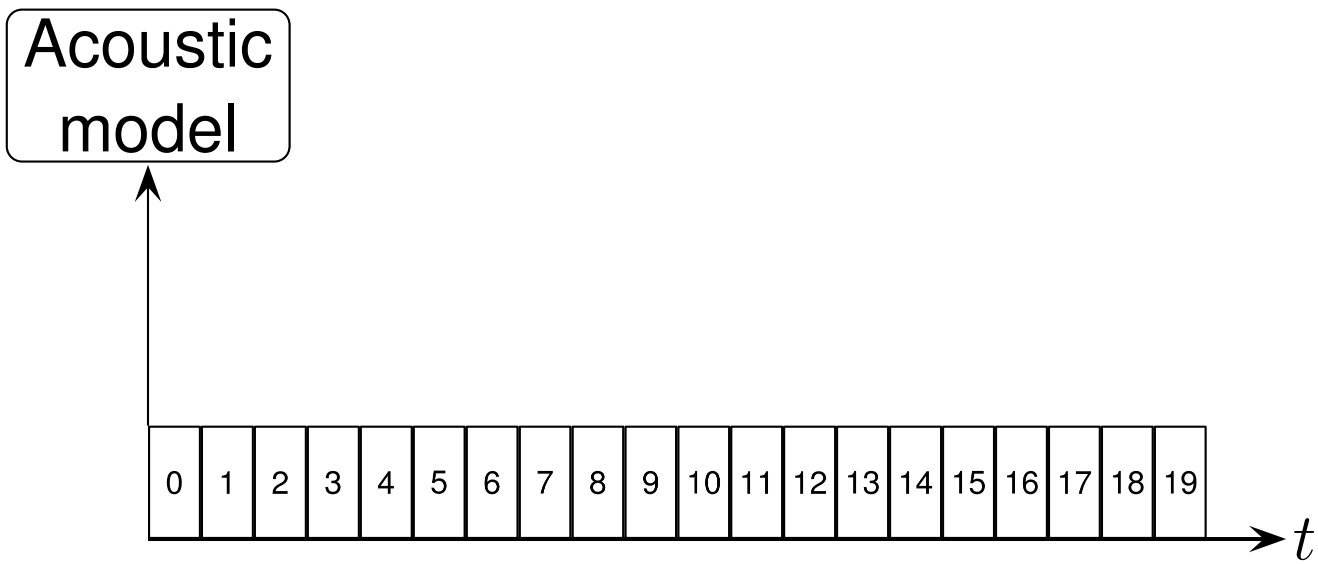
- Acoustic features normalization
- Limited future context for the acoustic signal
- Time constraints (Delay)

Proposed techniques:

- Running feature normalization
- Sliding window over the acoustic feat., similar to (Zeyer, 2016)
- Fast one-pass decoder

MLLP

One-pass decoder: off-line vs streaming



Normalization delay impact

- n_{norm} seconds gathering stats to normalize
- Lookahead window fixed to 0.5 seconds

n_{norm} (sec)	*LibriSpeech*	*TED-LIUMv3*
0	15.6	9.7
1	11.0	8.2
2	10.0	8.1
4	9.6	7.9
8	9.4	7.7
∞	9.4	7.6

Lookahead impact

- $n_{\text{lookahead}}$ seconds for the lookahead context window
- n_{norm} is fixed to 2 seconds

$n_{\text{lookahead}}$ (sec)	*LibriSpeech*	*TED-LIUMv3*
0.125	17.1	10.5
0.250	11.6	8.8
0.500	10.0	8.1
1.000	9.9	7.9
2.000	10.2	7.8

Lookahead/Normalization delay impact

- Expected latency in a fully-streaming regime
- Results in latency discarding the initial $n_{\text{norm}} = 2$ secs delay

	LibriSpeech		*TED-LIUMv3*	
$n_{\text{lookahead}}$ (sec)	WER	Latency (sec)	WER	Latency (sec)
0.125	17.1	0.6	10.5	0.3
0.250	11.6	0.6	8.8	0.5
0.500	10.0	0.8	8.1	0.8
1.000	9.9	1.4	7.9	1.3
2.000	10.2	2.9	7.8	2.3

MLLP

Off-line vs Streaming setup

- Final comparison between off-line vs streaming setup
- Streaming: $n_{\text{norm}} = 2$ secs, $n_{\text{lookahead}} = 0.5$ secs
- Off-line: The whole utterance available

	LibriSpeech	*TED-LIUMv3*
Off-line setup	10.2	8.2
Streaming setup	10.7	8.7

Further work

- Transformer LM [Baquero-Arnal et al., 2020]
- More results [Jorge et al., 2021]

MLLP

3 Bridging ASR and MT

Direct Segmentation Models for Streaming Speech Translation [Iranzo-Sánchez et al., 2020]

Text format mismatch

- ***ASR output:*** i declare resumed the session of the european parliament adjourned on friday seventeen december nineteen ninety-nine
- ***Standard MT input:*** I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999.
  - ▷ Preprocess MT source training data

Preprocessing	En-De	En-Es	En-Fr	Es-En	Fr-En	De-En
Conventional MT	22.4	28.0	23.4	26.5	25.4	21.3
Special ST	26.5	35.5	29.3	33.8	29.9	25.8

  - ▷ Repunctuation/recasing Neural model (upcoming work)

Sentence segmentation: Previous work

Acoustic-based segmentation Heuristics using acoustic info, i.e. Voice Activity Detection [Silvestre-Cerdà et al., 2012]

LM-based segmentation
Use LM to compute end-of-sentence (EOS) probability [Stolcke and Shriberg, 1996, Wang et al., 2016, 2019]

Monolingual MT segmentation
Translation adds punctuation marks, thus segmentation [Cho et al., 2012, 2015, 2017]

Statistical framework

- Sequence of words w_1^J to be split into non-overlapping chunks
- Sequence of split/non-split decisions c_1^J:

$$c_j = \begin{cases} 1 & \text{if word } w_j \text{ ends a chunk} \\ 0 & \text{otherwise} \end{cases}$$

- Optionally, sequence of word-based acoustic features $\check{x}_1^J$
- Optimal segmentation $\hat{c}_1^J$ according to

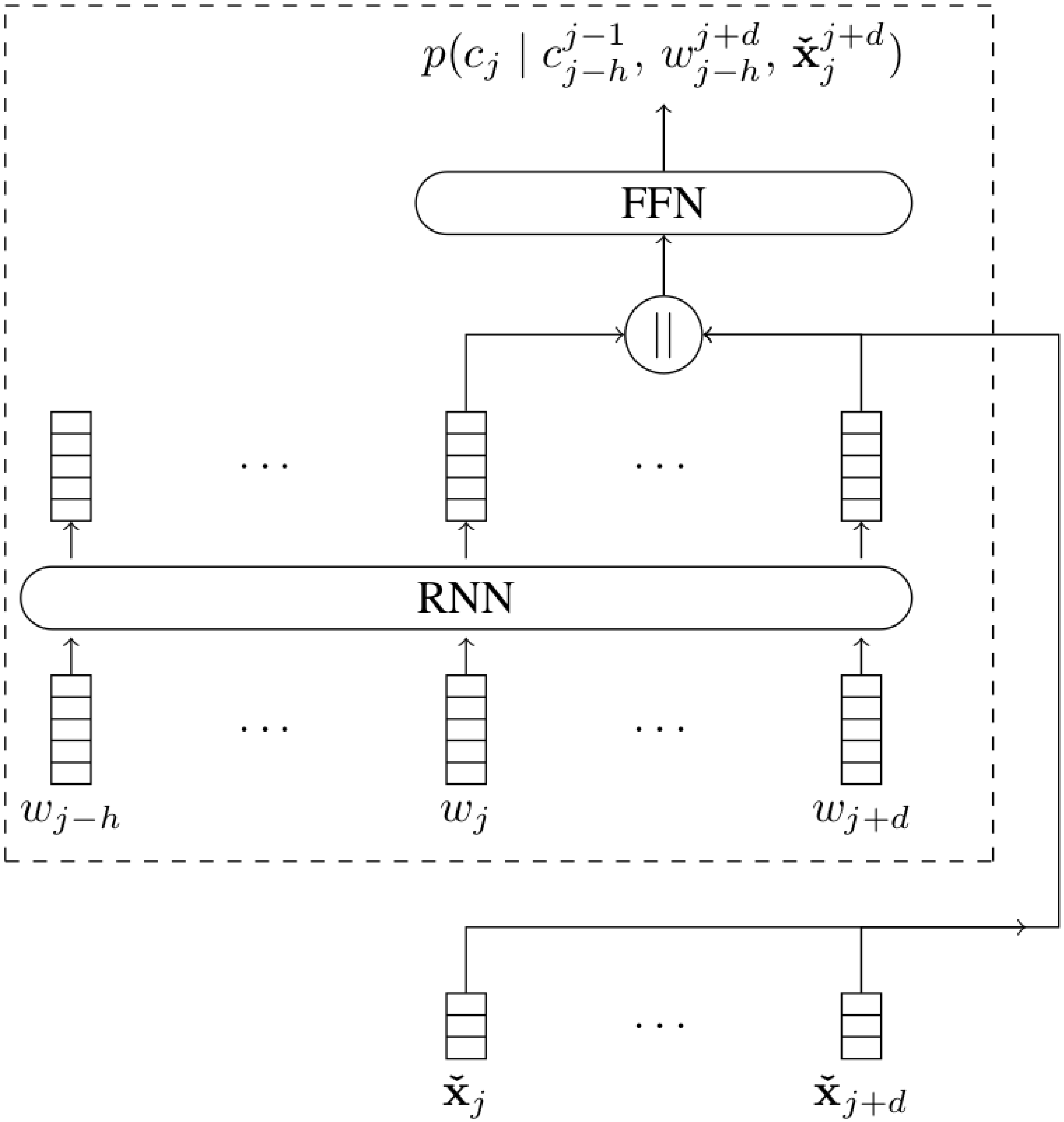
$$\begin{aligned} \hat{c}_1^J &= \underset{c_1^J}{\arg\max}\; p(c_1^J \mid w_1^J, \check{x}_1^J) \\ &= \underset{c_1^J}{\arg\max} \prod_{j=1}^{J} p(c_j \mid c_1^{j-1}, w_1^J, \check{x}_1^J) \end{aligned}$$

MLLP

Direct segmentation model

- Split decision at current word j only depends on
  - h words into the past (*history size*)
  - d words into the future (*length of future window*)
- Under these assumptions, our search problem is

$$\hat{c}_1^J = \operatorname*{arg\,max}_{c_1^J} \prod_{j=1}^{J} p(c_j \mid c_{j-h}^{j-1}, w_{j-h}^{j+d}, \check{x}_{j-h}^{j+d})$$


$p(c_j \mid c_{j-h}^{j-1}, w_{j-h}^{j+d}, \check{\mathbf{x}}_j^{j+d})$
FFN
||
RNN
w_{j-h}
w_j
w_{j+d}
$\check{\mathbf{x}}_j$
$\check{\mathbf{x}}_{j+d}$

ASR input

Segmenter	En-De	En-Es	En-Fr	Fr-En	De-En	Es-En
Baseline (VAD)	26.5	35.5	29.3	29.9	25.8	33.8
Text	27.6	37.0	29.4	31.6	28.1	34.7
Audio w/o RNN	28.4	37.2	30.0	32.1	28.3	34.4
Audio w/ RNN	28.4	37.3	30.1	32.1	28.2	33.9
Oracle	31.6	41.3	33.6	35.3	31.3	38.1

- Audio models improve over Text models
- Larger w (4) and h (10) tend to perform better

4 Streaming MT: Evaluation

Stream-level Latency Evaluation for Simultaneous Machine Translation [Iranzo-Sánchez et al., 2021]

Preliminars: Simultaneous MT

(Sentence-level) Simultaneous Machine Translation

- Incrementally translate a sentence before it is fully available

- For every sentence pair $(\boldsymbol{x}, \boldsymbol{y})$,

$$\hat{y}_i = \underset{y \in \mathcal{Y}}{\arg\max}\, p\left(y \mid x_1^{g(i)}, y_1^{i-1}\right)$$

- Delay function $g(i)$: # src. words available for writing i-th word.

Simultaneous MT models

- A simultaneous MT model is characterized by its policy

- At each timestep, the policy decides between 2 actions:
  - READ a word (wait for more context)
  - WRITE a word

- Baseline policy: Wait-k translation
  - First wait for k words to arrive (READ),
  - then alternate between WRITE and READ

$$g(i) = \left\lfloor k + \frac{i-1}{\gamma} \right\rfloor$$

Simultaneous MT Evaluation

Latency for the n-th sentence pair

$$L(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n) = \frac{1}{Z(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)} \sum_i C_i(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)$$

- Z: Normalisation function
- C_i a cost function for each target position i

Latency for the evaluation set

- Average of the latencies of each sentence pair

Cost function

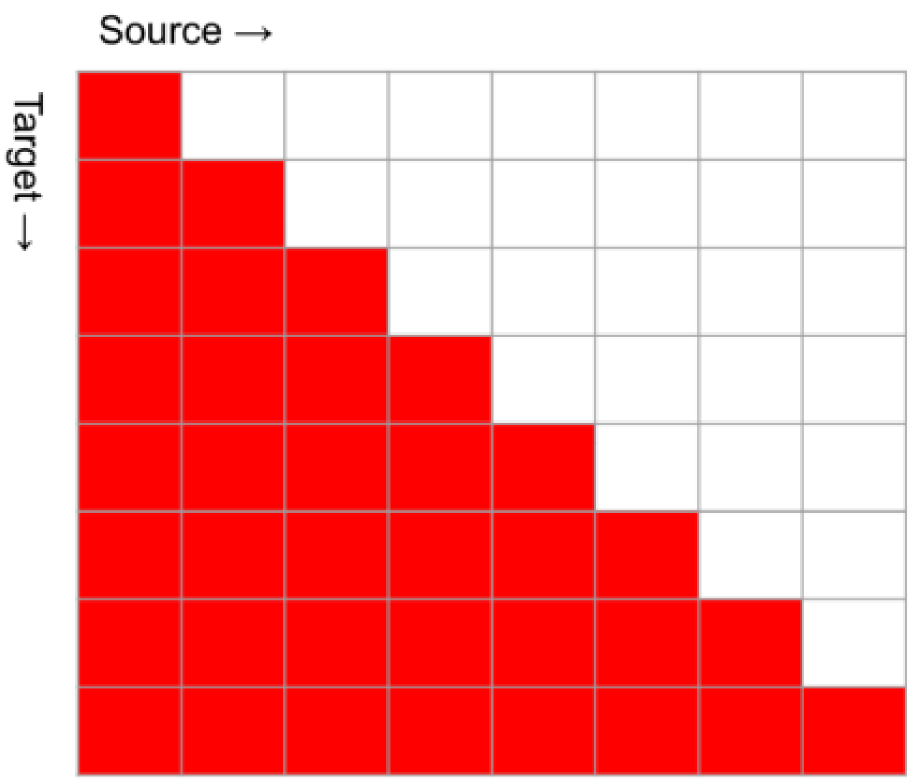
$$C_i(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \begin{cases} g(i) & \text{AP} \\ g(i) - \frac{i-1}{\gamma_n} & \text{AL} \\ g'(i) - \frac{i-1}{\gamma_n} & \text{DAL} \end{cases} \tag{1}$$

Normalisation function

$$Z(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \begin{cases} |\boldsymbol{x}| \cdot |\hat{\boldsymbol{y}}| & \text{AP} \\ \underset{i : g(i) = |\boldsymbol{x}|}{\arg\min} \; i & \text{AL} \\ |\hat{\boldsymbol{y}}| & \text{DAL} \end{cases} \tag{2}$$

Average Proportion (AP)

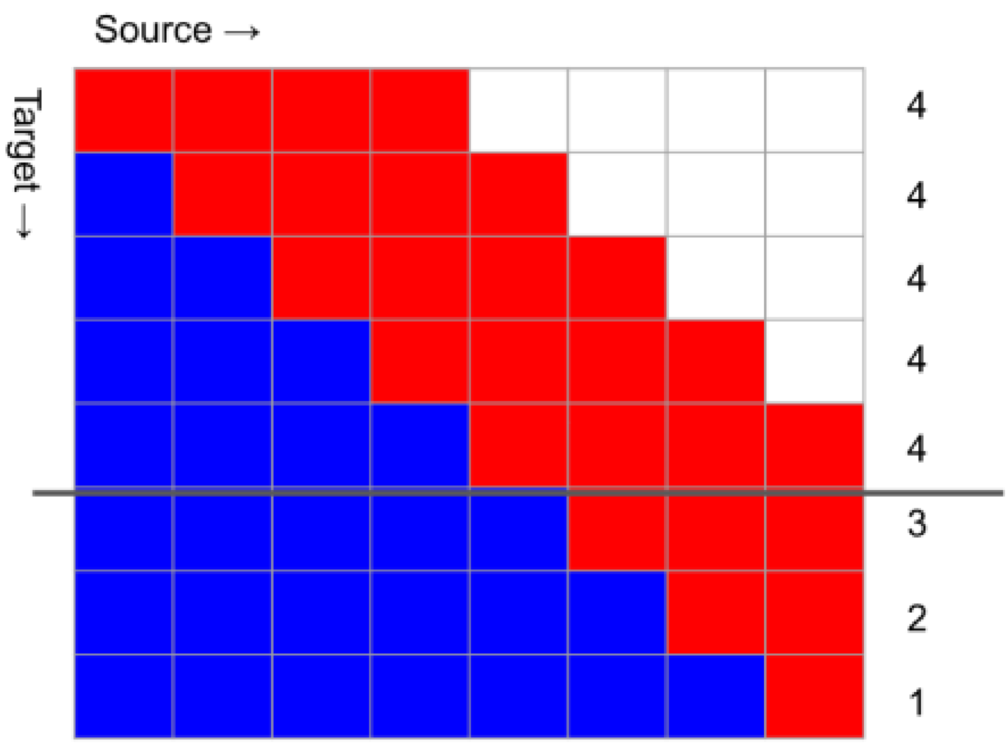
$$L(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n) = \frac{1}{|\boldsymbol{x}| \cdot |\hat{\boldsymbol{y}}|} \sum_i g(i)$$



(Image source: [Huang et al., 2020])

Average Lagging (AL)

$$L(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n) = \frac{1}{\underset{i:g(i)=|\boldsymbol{x}|}{\arg\min}\ i} \sum_i g(i) - \frac{i-1}{\gamma_n}$$

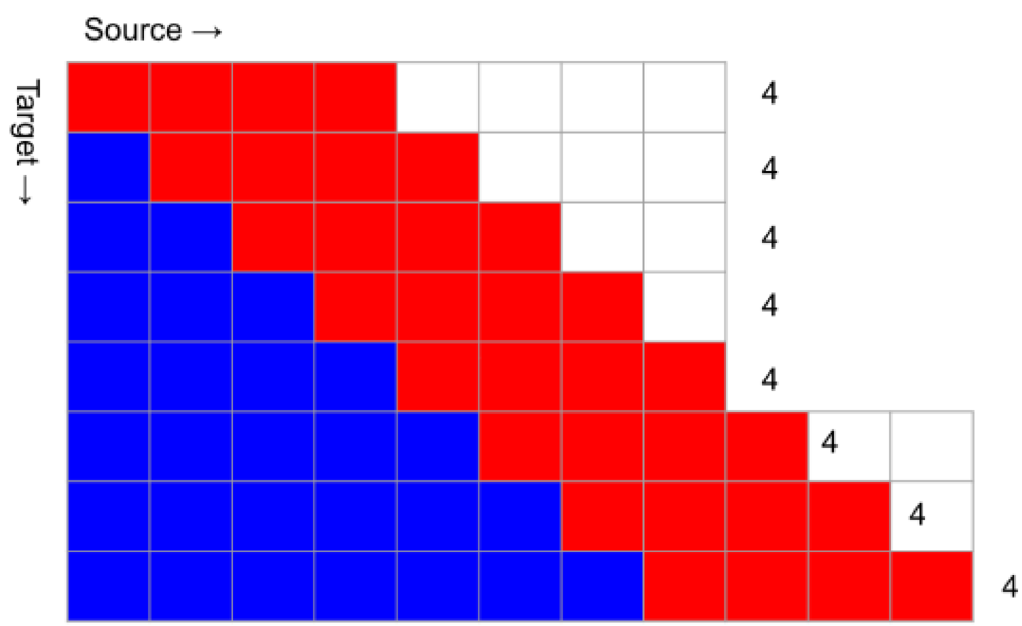


(Image source: [Huang et al., 2020])

Differentiable Average Lagging (DAL)

$$L(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n) = \frac{1}{|\hat{\boldsymbol{y}}|} \sum_i g'(i) - \frac{i-1}{\gamma_n}$$

$$g'(i) = \max \begin{cases} g(i) \\ g'(i-1) + \frac{1}{\gamma} \end{cases} \tag{3}$$



(Image source: [Huang et al., 2020])

Simultaneous Translation Evaluation

Downside

- Sentences are evaluated in isolation
- Fixed segmentation must be used to compare systems
- Unrealistic scenario

Proposed approach

- Evaluate latency of the entire stream

Simultaneous Translation Evaluation: Previous work

Concat-1[Schneider and Waibel, 2020]

- Concat all text into a single sentence, translate & evaluate

Drawbacks

- This assumes a constant writing rate (γ) for the entire stream
- Is this realistic?

Concat 1 - Example

- $|\boldsymbol{x}_1| = 2$, $|\hat{y}_1| = 2$, $|\gamma_1| = 1$
- $|\boldsymbol{x}_2| = 2$, $|\hat{y}_2| = 4$, $|\gamma_2| = 2$

									L
Concat-1		i	1	2	3	4	5	6	
		$g(i)$	1	2	3	3	4	4	
		$\frac{i-1}{\gamma}$	0	0.6	1.3	2.0	2.6	3.3	
	C_i	AP	1	2	3	3	4	4	0.7
		AL	1	1.3	1.6	1	1.3	-	1.2
		DAL	1	1.3	1.6	1.6	1.6	1.6	1.5
Ind. Sent.		i	1	2	1	2	3	4	
		$g(i)$	1	2	1	1	2	2	
		$\frac{i-1}{\gamma}$	0.0	1.0	0.0	0.5	1.0	1.5	
	C_i	AP	1	2	1	1	2	2	0.8
		AL	1	1	1	0.5	1	-	0.9
		DAL	1	1	1	1	1	1	1.0

MLLP

Concat 1 - Cont.

- AP $\rightarrow 0.5$
- AL and DAL do not reflect real behaviour of the model
  - Oracle writing speed is always under/over-estimated
- DAL grows larger and larger
- Streaming evaluation is unfeasible with a single, fixed γ

Our proposal

- Key idea: Need local (sentence-level) estimation of γ, γ_n
- Keep track of latency with a global delay, $G(s)$
- Convert to local representation and compared with local oracle

MLLP

Evaluation methods

$G(s)$: # stream src words available for writing s-th tgt stream word

$$C_i(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n) = \begin{cases} g_n(i) & \text{AP} \\ g_n(i) - \frac{i-1}{\gamma_n} & \text{AL} \\ g'_n(i) - \frac{i-1}{\gamma_n} & \text{DAL} \end{cases}$$

$$\underbrace{g_n(i)}_{\text{Local delay}} = \underbrace{G(i + |\hat{\boldsymbol{y}}_1^{n-1}|)}_{\text{Global delay}} - \underbrace{|\boldsymbol{x}_1^{n-1}|}_{\text{Local operator}}$$

Evaluation methods

$$g'_n(i) \quad \max \begin{cases} g_n(i) \\ \begin{cases} g'_{n-1}(|\boldsymbol{x}_{n-1}|) + \frac{1}{\gamma_{n-1}} & i = 1 \\ g'_n(i-1) + \frac{1}{\gamma_n} & i > 1 \end{cases} \end{cases}$$

Segmentation

- We need sentence-level alignment from a stream translation Y
- Do as for quality evaluation: Re-align sentences
  - Minimum edit distance: MWER [Matusov et al., 2005]
- After re-alignment, we obtain pairs $(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)$

MLLP

AL Results

- Train data: IWSLT2020 En$\leftrightarrow$De except OpenSubtitles

- Eval data: IWSLT2010 De$\rightarrow$En

- 1 system + 3 oracles:
  - Real: DS segmenter + Wait-k system
  - + In. Seg: Use ref. input segmentation instead of DS
  - + Out. Seg: Use ref. output segmentation instead of MWER
  - + Policy : Use oracle γ_n for each sentence

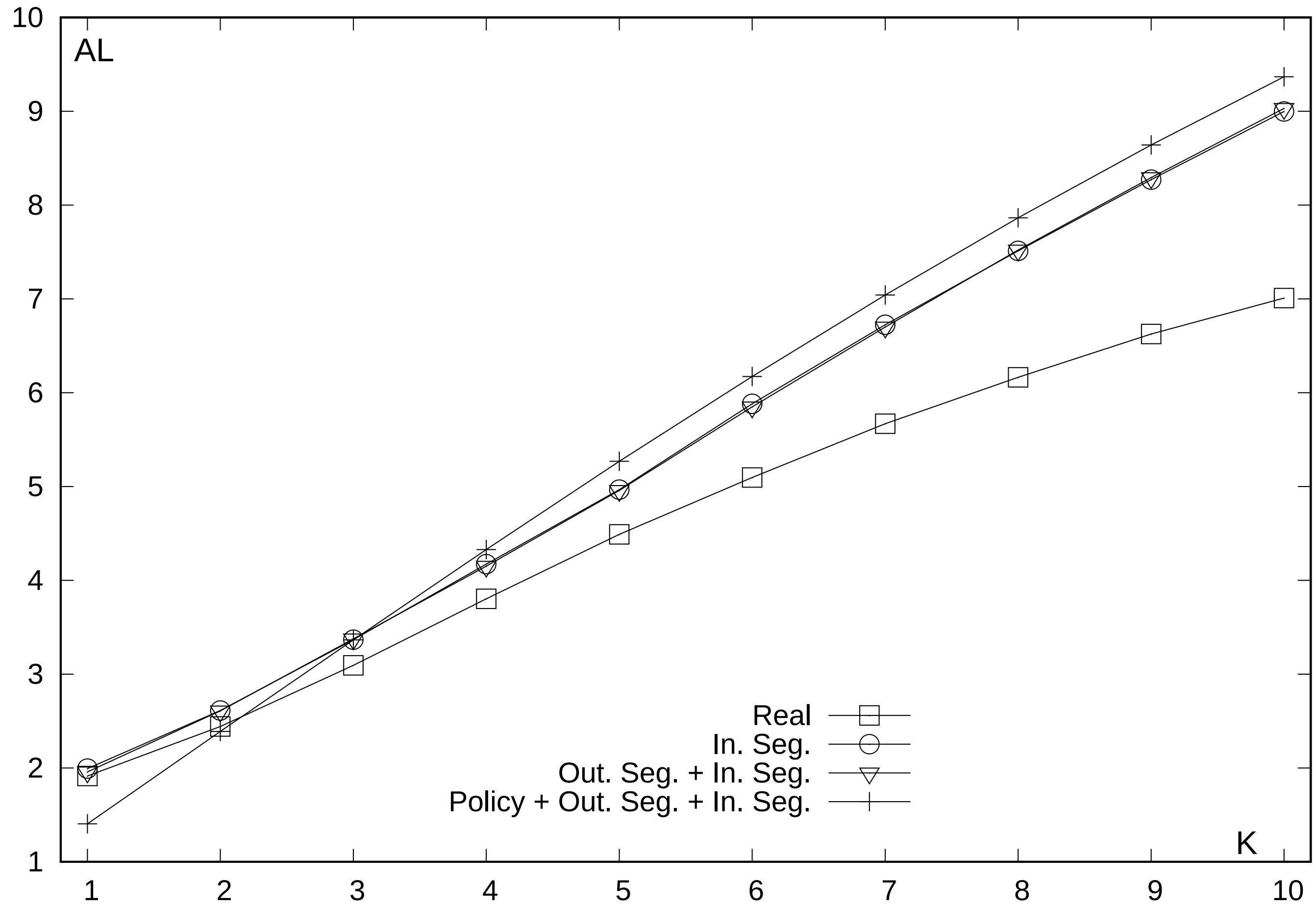
AL Results

Concat-1

	Wait-k				
System	1	2	3	4	5
Real	-9.7	-12.0	-45.2	-23.7	-8.5
+In. Seg.	-42.9	-29.0	17.4	-10.1	25.5
+ Policy	14.2	15.1	16.0	16.8	17.6

AL Results(cont.)

Proposed approach



- Results ranked by increasing order of k
- Interpretable and accurate results

5 Streaming MT: Models & Baseline

From Simultaneous to Streaming Machine Translation by Leveraging Streaming History [Iranzo-Sánchez et al., 2022]

- Translate an input stream X into a target stream Y
- Global delay $G(i)$

$$\hat{Y}_i = \underset{y \in \mathcal{Y}}{\arg\max}\, p\left(y \mid X_1^{G(i)}, Y_1^{i-1}\right)$$

- For efficiency, we introduce the history function $H(i)$

$$\hat{Y}_i = \underset{y \in \mathcal{Y}}{\arg\max}\, p\left(y \mid X_{G(i)-H(i)+1}^{G(i)}, Y_{i-H(i)}^{i-1}\right)$$

MLLP

Streaming MT Baseline

Segmentation

- a_n: Starting position of n-th source sentence
- b_n: Starting position of n-th target sentence
- $|\mathbf{a}| = |\mathbf{b}| = N$

Streaming MT Baseline

Policy

- Simultaneous (sentence-level) wait-k:

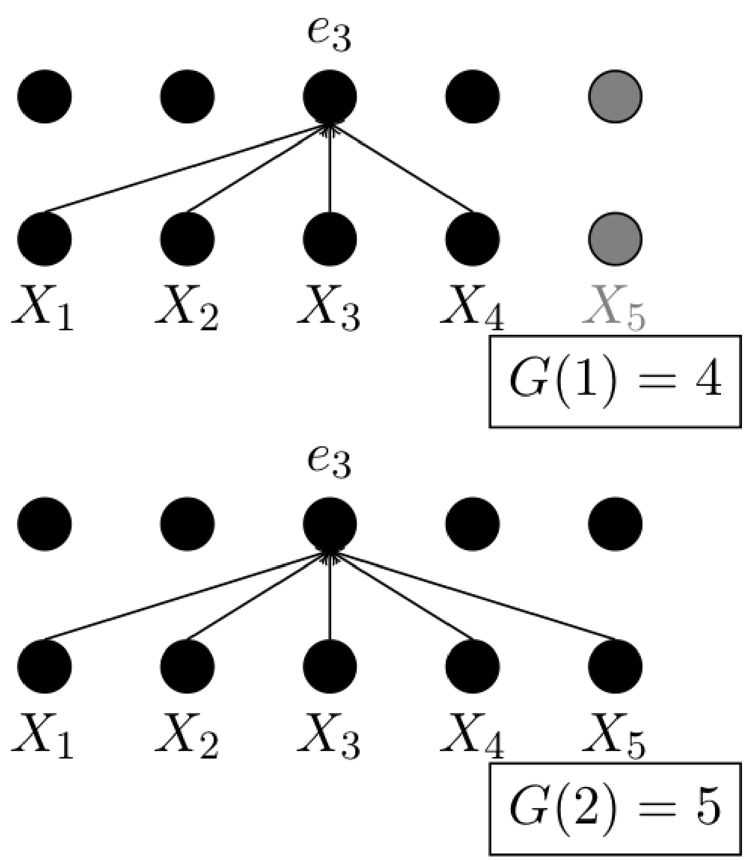
$$g(i) = \left\lfloor k + \frac{i-1}{\gamma} \right\rfloor$$

- Streaming MT wait-k:

$$G(i) = \left\lfloor k + \frac{i-b_n}{\gamma} \right\rfloor + a_n - 1$$

- $b_n \leq i < b_{n+1}$

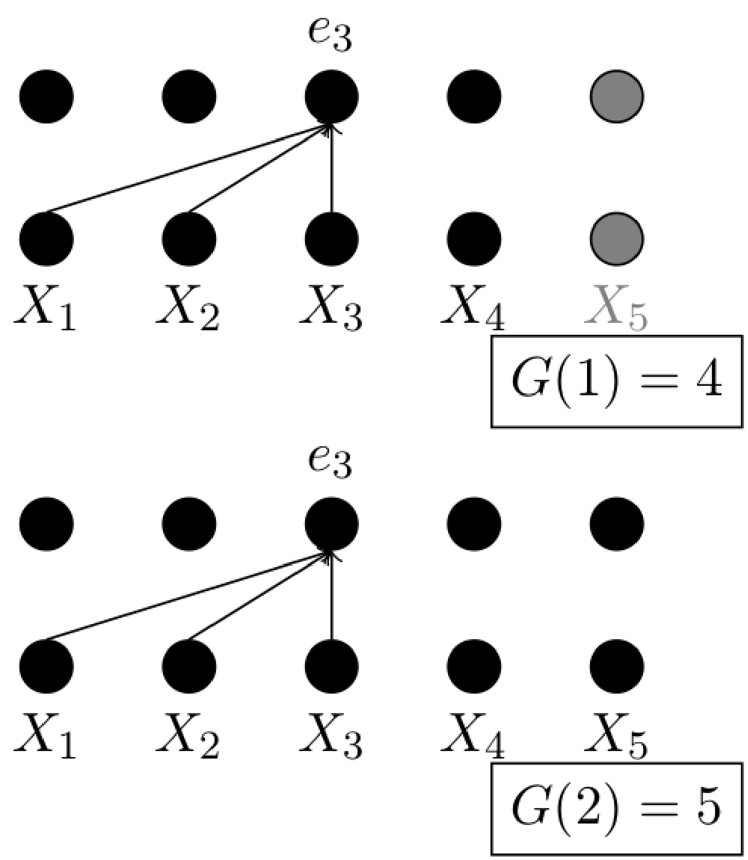
Streaming MT Baseline: Encoders



Bidirectional - Standard MT

$$e_j^{(l)} = \mathrm{Enc}\left(e^{(l-1)}_{G(i)-H(i)+1:G(i)}\right)$$

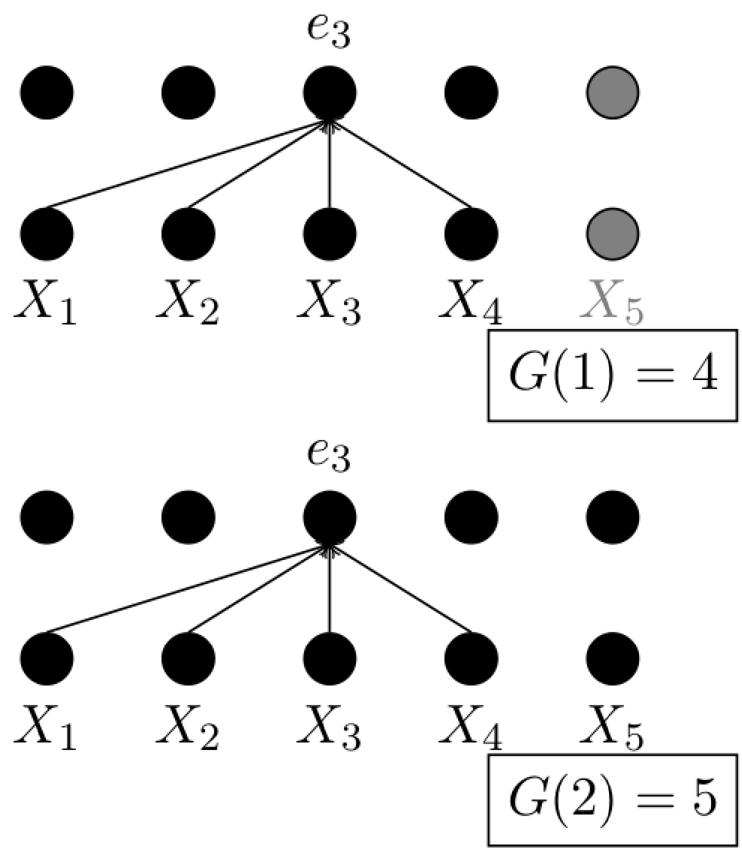
Streaming MT Baseline: Encoders



- Unidirectional - [Ma et al., 2019, Elbayad et al., 2020]

$$e_j^{(l)} = \mathrm{Enc}\left(e^{(l-1)}_{G(i)-H(i)+1:j}\right)$$

Streaming MT Baseline: Encoders



Partial Bidirectional Encoder (PBE) - This work

$$e_j^{(l)} = \mathrm{Enc}\left(e^{(l-1)}_{G(i)-H(i)+1:\max(G(i)-H(i)+k,j)}\right)$$

Streaming MT Baseline: System training

Sentence pair	Source	Target
1	$x_{1,1}\ x_{1,2}$	$y_{1,1}\ y_{1,2}$
2	$x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}$	$y_{2,1}\ y_{2,2}\ y_{2,3}$
3	$x_{3,1}\ x_{3,2}\ x_{3,3}$	$y_{3,1}\ y_{3,2}\ y_{3,3}$

Sample	Source
1	[DOC] $x_{1,1}\ x_{1,2}$ [BRK]
2	[DOC] $x_{1,1}\ x_{1,2}$ [SEP] $x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}$ [BRK]
3	[CONT] $x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}$ [SEP] $x_{3,1}\ x_{3,2}\ x_{3,3}$ [BRK]

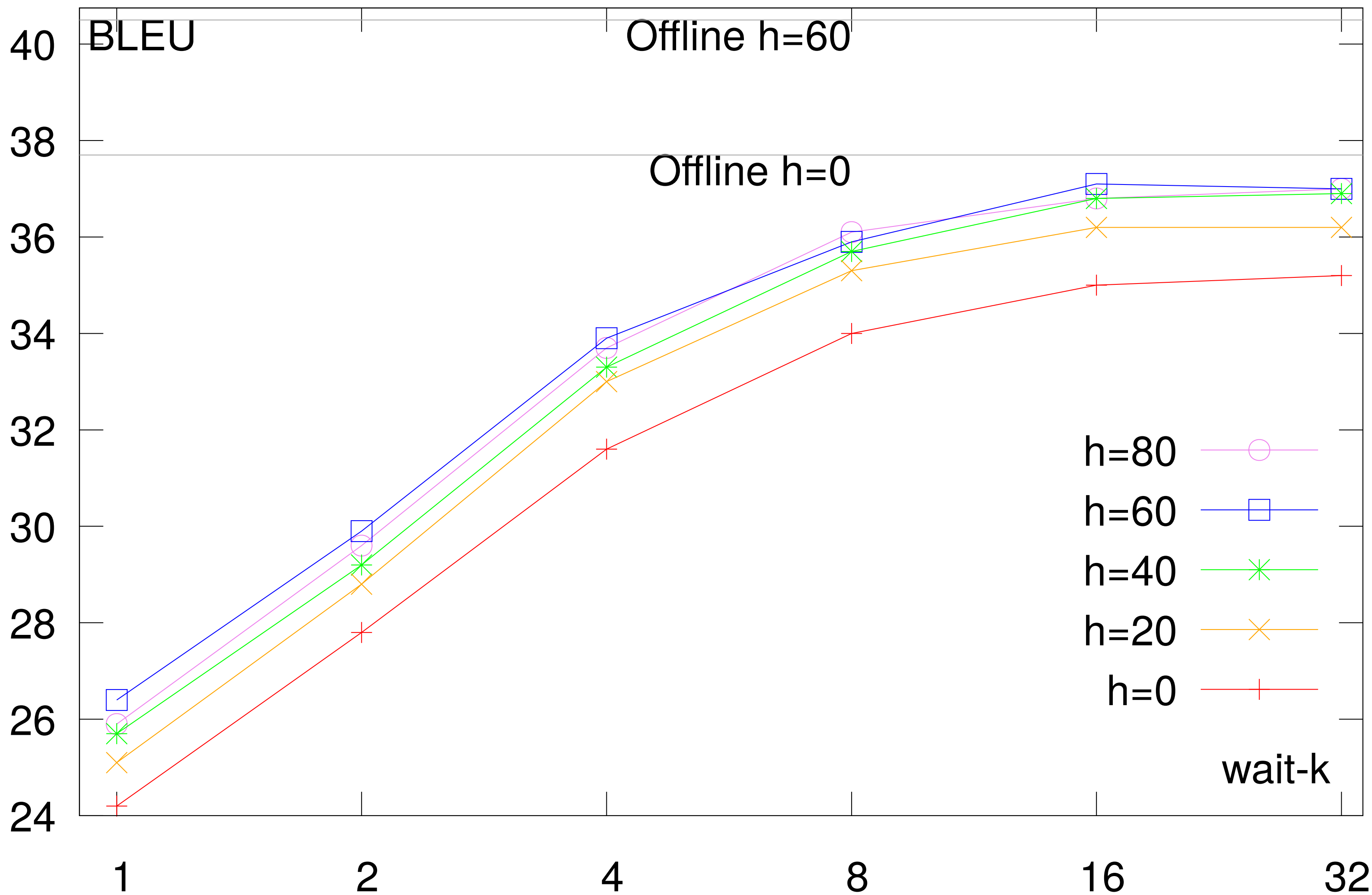
Sample	Target
1	[DOC] $y_{1,1}\ y_{1,2}$ [BRK]
2	[DOC] $y_{1,1}\ y_{1,2}$ [SEP] $y_{2,1}\ y_{2,2}\ y_{2,3}$ [BRK]
3	[CONT] $y_{2,1}\ y_{2,2}\ y_{2,3}$ [SEP] $y_{3,1}\ y_{3,2}\ y_{3,3}$ [BRK]

Experiments: Setup

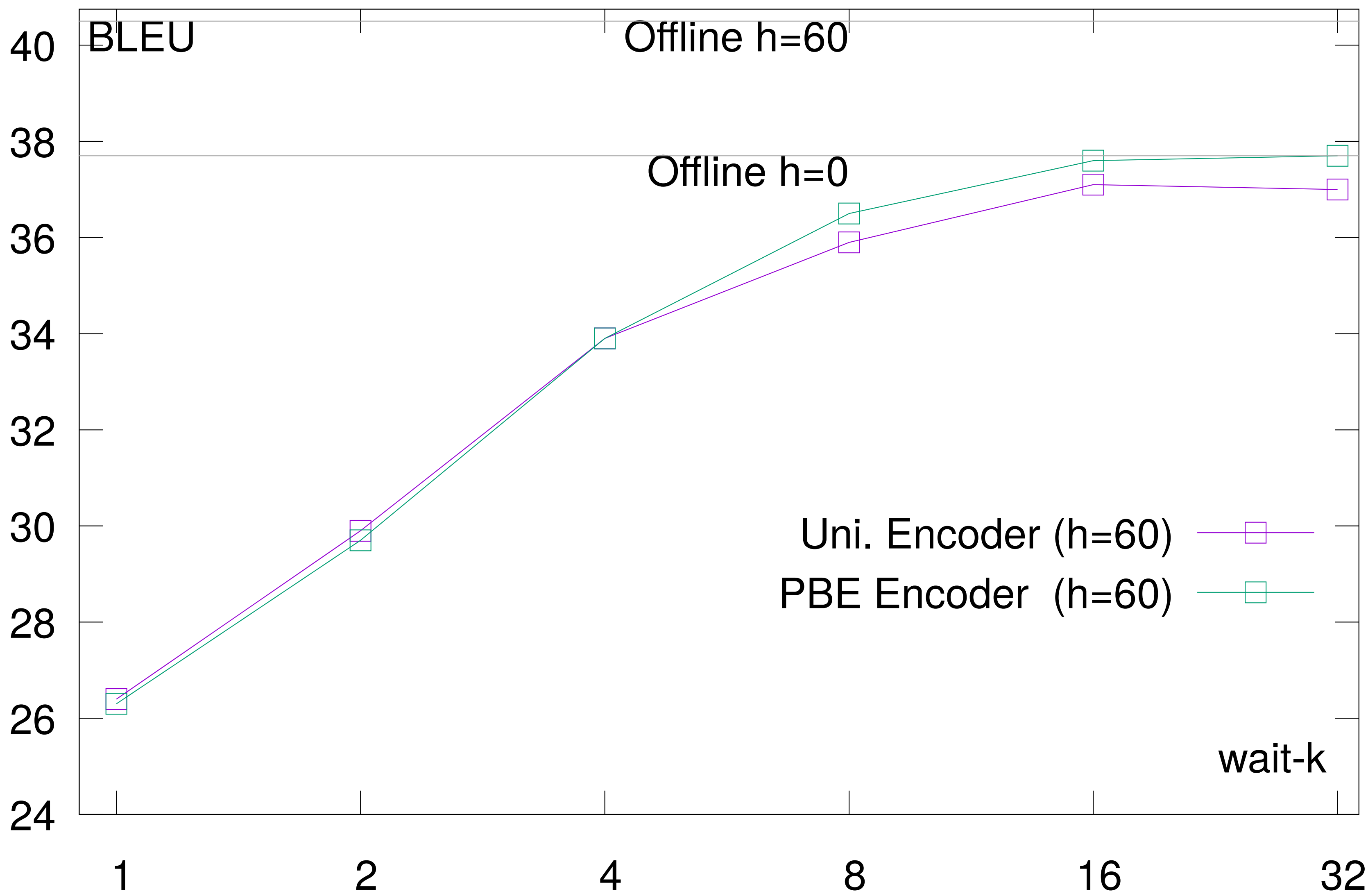
- Train data: IWSLT2020 En$\leftrightarrow$De except OpenSubtitles
- Eval data
  - IWSLT2010 De$\rightarrow$En
  - IWSLT2020 En$\rightarrow$De
- Finetune on MuST-C train
  - Same setup as [Schneider and Waibel, 2020]
- Transformer Big model, 40k BPE subwords
- Multi-path wait-k policy [Elbayad et al., 2020]

Experiments: Streaming history size

IWSLT 2010 Dev (De $\rightarrow$ En)



Experiments: PBE Encoder



Experiments: Comparison with SoTA

Streaming MT, IWSLT 2010

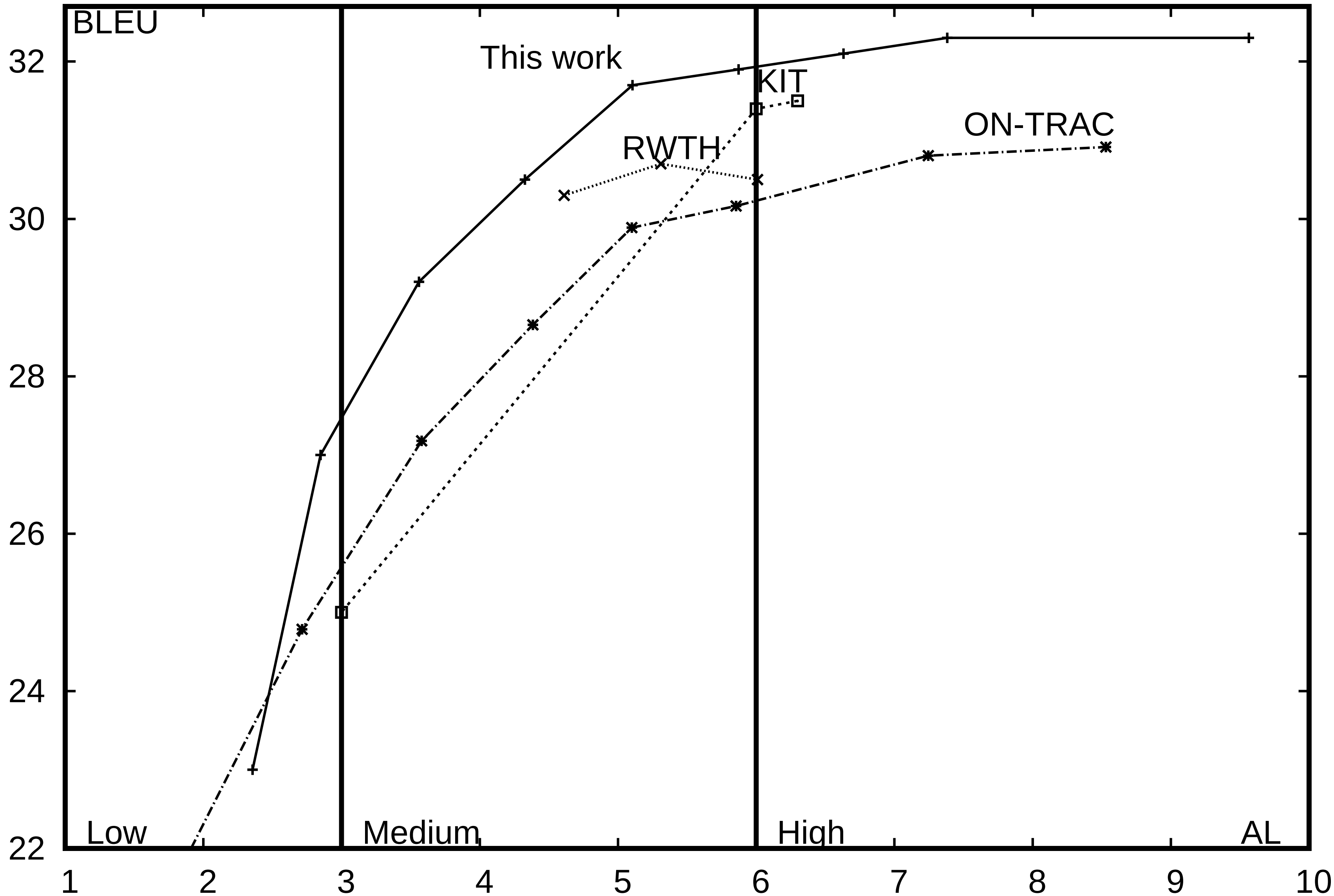
Model	BLEU	AP	AL	DAL
[Schneider and Waibel, 2020]	30.3	10.3	100.1	101.8
This work	29.5	1.2	11.2	17.8

- Latency measured with streaming AP/AL/DAL [Iranzo-Sánchez et al., 2021]
- Similar performance with a fraction of the latency
- Adaptive policy of ACT falls behind (no catch-up mechanism)
- Wait-k + segmenter ensure model keeps up with the speaker

MLLP

Experiments: Comparison with SoTA

IWSLT 2020: MuST-C tst-COMMON



Thanks for your attention!

Full details available in the papers

Code for segmenter/MT: https://github.com/jairsan

References

Pau Baquero-Arnal, Javier Jorge, Adrià Giménez, Joan Albert Silvestre-Cerdà, Javier Iranzo-Sánchez, Albert Sanchis, Jorge Civera, and Alfons Juan. Improved Hybrid Streaming ASR with Transformer Language Models. In *Proc. of Interspeech*, pages 2127–2131, 2020.

Eunah Cho, Jan Niehues, and Alex Waibel. Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In *Proc. of IWSLT*. ISCA, 2012.

Eunah Cho, Jan Niehues, Kevin Kilgour, and Alex Waibel. Punctuation insertion for real-time spoken language translation. In *Proc. of IWSLT*. ISCA, 2015.

Eunah Cho, Jan Niehues, and Alex Waibel. NMT-Based Segmentation and Punctuation Insertion for Real-Time Spoken Language Translation. In *Proc. of Interspeech*, pages 2645–2649. ISCA, 2017.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. of Interspeech*, pages 1461–1465, 2020.

Liang Huang, Colin Cherry, Mingbo Ma, Naveen Arivazhagan, and Zhongjun He. Simultaneous translation. In *Proc. of EMNLP: Tutorial Abstracts*, pages 34–36. Association for Computational Linguistics, 2020.

Javier Iranzo-Sánchez, Adrià Giménez, Joan Albert Silvestre-Cerdà, Pau Baquero, Jorge Civera, and Alfons Juan. Direct Segmentation Models for Streaming Speech Translation. In *Proc. of EMNLP*, pages 2599–2611. ACL, 2020.

Javier Iranzo-Sánchez, Jorge Civera Saiz, and Alfons Juan. Stream-level latency evaluation for simultaneous machine translation. In *Findings of ACL: EMNLP*, pages 664–670. ACL, 2021.

Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. From simultaneous to streaming machine translation by leveraging streaming history. In *Proc. of ACL*, 2022.

Javier Jorge, Adrià Giménez, Javier Iranzo-Sánchez, Jorge Civera, Albert Sanchis, and Alfons Juan. Real-Time One-Pass Decoder for Speech Recognition Using LSTM Language Models. In *Proc. of Interspeech*, pages 3820–3824, 2019.

Javier Jorge, Adrià Giménez, Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Jorge Civera, Albert Sanchis, and Alfons Juan. LSTM-Based One-Pass Decoder for Low-Latency Streaming. In *Proc. of ICASSP*, pages 7814–7818, 2020.

Javier Jorge, Adrià Giménez, Joan Albert Silvestre-Cerdà, Jorge Civera, Albert Sanchis, and Juan Alfons. Live streaming speech recognition using deep bidirectional lstm acoustic models and interpolated language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:148–161, 2021. doi: 10.1109/TASLP.2021.3133216.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proc. of ACL*, pages 3025–3036. ACL, 2019.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. Evaluating machine translation output with automatic sentence segmentation. In *Proc. of IWSLT*. ISCA, 2005.

Felix Schneider and Alexander Waibel. Towards stream translation: Adaptive computation time for simultaneous machine translation. In *Proc. of IWSLT*, pages 228–236. ACL, 2020.

Joan Albert Silvestre-Cerdà, Adrià Giménez, Jesús Andrés-Ferrer, Jorge Civera, and Alfons Juan. Albayzin Evaluation: The PRHLT-UPV Audio Segmentation System. In *Proc. of IberSPEECH 2012*, pages 596–600, 2012.

Andreas Stolcke and Elizabeth Shriberg. Automatic linguistic segmentation of conversational speech. In *Proc. of ICSLP*, volume 2, pages 1005–1008. ISCA, 1996.

Xiaolin Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. An Efficient and Effective Online Sentence Segmenter for Simultaneous Interpretation. In *Proc. of WAT*, pages 139–148. The COLING 2016 Organizing Committee, 2016.

Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. Online Sentence Segmentation for Simultaneous Interpretation using Multi-Shifted Recurrent Neural Network. In *Proc. of MT Summit XVII Volume 1: Research Track*, pages 1–11. EAMT, 2019.

MLLP