Online Neural Machine Translation with 2D Convolutions

Maha Elbayad Laurent Besacier Jakob Verbeek

Under review at EMNLP 2019

July 5, 2019

Outline

- Low latency translation.
- 2D convolutional architecture for MT.
- 2D convolutional architecture for low latency MT.
- Deterministic and controllable latency.
- Experimental results.
- Conclusion.

Latency of sequence-to-sequence models

Seq2seq: a conditional language model that assigns probabilities to a sequence of tokens $\boldsymbol{y} = (y_1, ..., y_{|\boldsymbol{y}|})$ given a conditioning sequence $\boldsymbol{x} = (x_1, ..., x_{|\boldsymbol{x}|})$.

$$\log p_\theta(\mathbf{y}|\mathbf{x}) = \sum_{t=1}^{|y|} \log p_\theta(y_t|\mathbf{x}, \boldsymbol{y}_{<t})$$

The model's parameters optimized through MLE:

$$\mathcal{L} = -\log p_\theta(\boldsymbol{y}|\boldsymbol{x})$$

Latency: stems from $y_t|\boldsymbol{x}$ as we have to read the full source $\boldsymbol{x}$.

Latency of sequence-to-sequence models

Motivation: build a seq2seq model for online / as-you-type translation.

Introduce a hidden variable z_t as the length of the context read by timestep t:

$$
\begin{aligned}
p_\theta(\boldsymbol{y}|\boldsymbol{x}) &= \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}) \\
&= \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z}) p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x}) \\
p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x}) &= \prod_t p_\theta(y_t|\boldsymbol{x}_{\leq z_t}, \boldsymbol{y}_{<t})
\end{aligned}
$$

Latency of sequence-to-sequence models

Two key points:

1. Encode $y_t | x_{\leq z_t}, y_{<t}$

$$p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x}) = \prod_t p_\theta(y_t|\boldsymbol{x}_{\leq z_t}, \boldsymbol{y}_{<t})$$

2. Model the hidden variables $\boldsymbol{z} = (z_1, \ldots, z_{|\boldsymbol{y}|})$.

Latency of sequence-to-sequence models

1. Encode $y_t | x_{\leq z_t}, y_{<t}$

$$p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x}) = \prod_t p_\theta(y_t|\boldsymbol{x}_{\leq z_t}, \boldsymbol{y}_{<t})$$

With attention-based encoder-decoder architectures we need to re-score the source for every context size z_t.
With $(h_1, \ldots h_{|\boldsymbol{y}|})$ the decoder states and $(s_1, \ldots, s_{|\boldsymbol{x}|})$ the encoder states (both evaluated autoregressively):

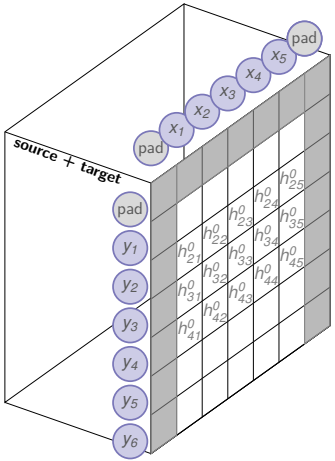
$$\boldsymbol{\alpha}_{z_t} = \text{softmax}(e_i)_{1 \leq i \leq z_t}$$
$$e_i = \text{score}(h_t, s_i), \ \forall i$$

Latency of sequence-to-sequence models

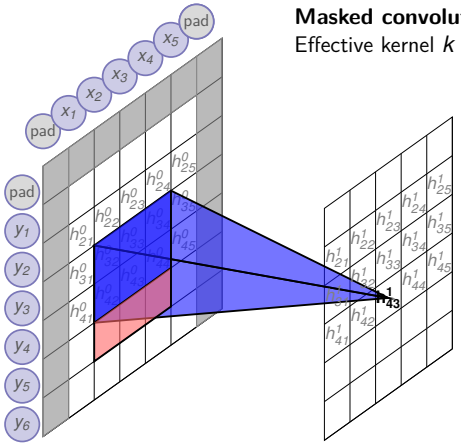
1. Encode $y_t | x_{\leq z_t}, y_{<t}$

$$p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x}) = \prod_t p_\theta(y_t|\boldsymbol{x}_{\leq z_t}, \boldsymbol{y}_{<t})$$

Leverage our 2D convolutional architecture where target encodings with different context sizes are readily available.

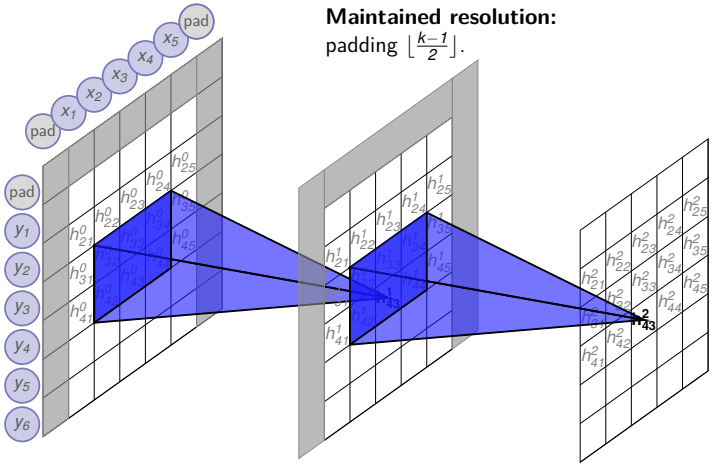


Input grid:
$\forall i, j \quad h^0_{ij} = concat(\text{embed}(y_i), \text{embed}(x_j))$
embed lookup tables for the source and target vocabularies.
+**Padding:** to maintain the input resolution.

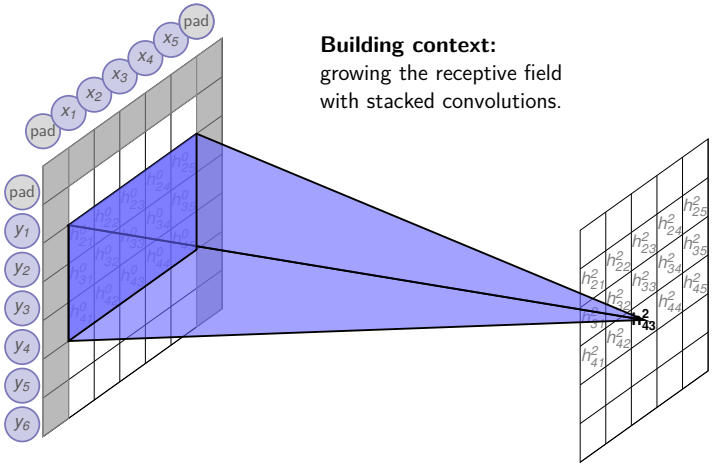


Masked convolution

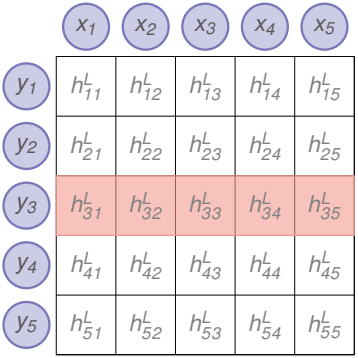
Effective kernel $k \times (1 + \lfloor \frac{k-1}{2} \rfloor)$


Maintained resolution:
padding $\lfloor \frac{k-1}{2} \rfloor$.
pad
pad
x_1
x_2
x_3
x_4
x_5
pad
y_1
y_2
y_3
y_4
y_5
y_6

Pervasive attention | 2D convolutional seq2seq model



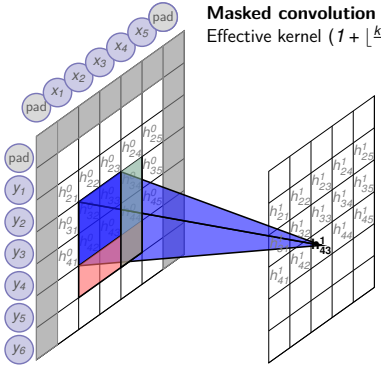
Building context:
growing the receptive field with stacked convolutions.



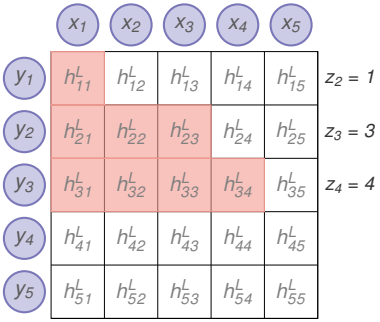
After L convolutional blocks:

- For each timestep t we have $\boldsymbol{x}$ representations.
- To model $y_t|\boldsymbol{x}$ we need a fixed size representation.
- A basic aggregation: $h_t = \text{Max-pool}(h_j)_{1 \leq j \leq |\boldsymbol{x}|}$

- Auto-regressive encoding of the source sequence with masked convolutions.



- Progressive pooling of the hidden states.



Max-pooling the states up to z_{t+1}.

$$h_t = \text{Max-pool}(h_{t,\leq z_{t+1}}).$$

The state is fed to a classifier to predict the next token:

$$p_\theta(y_{t+1}|y_{\leq t}, \boldsymbol{x}_{\leq z_{t+1}}) = \text{softmax}(Wh_t)$$

Less effective alternatives:

- Without pooling: $h_t = h_{t,z_t}$.
- Grid pooling: $h_t = \text{Max-pool}(h_{\leq t,\leq z_t})$.

Latency of sequence-to-sequence models

Two key points:

1. Encode $y_t | x_{\leq z_t}, y_{<t}$
2. Model the hidden variables $\boldsymbol{z} = (z_1, \ldots, z_{|\boldsymbol{y}|})$.

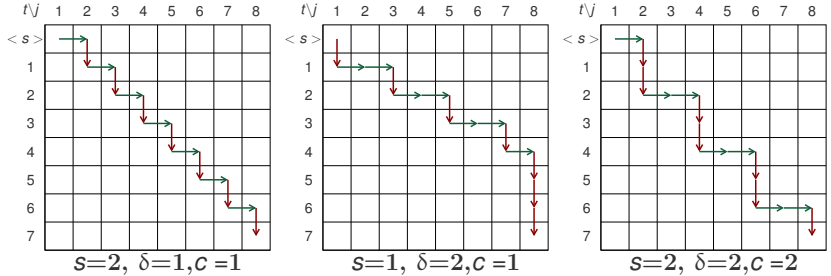
Two modelizations of $\boldsymbol{z}$ through a *'Controller'*:

- Deterministic ($\boldsymbol{z} \perp\!\!\!\perp \boldsymbol{x}, \boldsymbol{y}$) following a pre-defined schedule.
- Dynamic with $\boldsymbol{z}$ the latent variables of an HMM.

Deterministic z

$$z_t = s + \lfloor \frac{t-1}{c} \rfloor \delta$$

- Shift (s): context size at $t = 1$.
- Stepsize (δ): consecutive reads.
- Catchup (c): consecutive writes.



Two possible options for incorporating the variable z in the model:

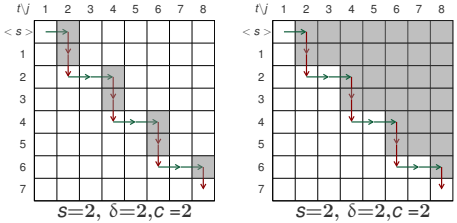
- **At inference:**
  With a model trained to generate after reading the full x (Wait-until-end: WUE), force the model to emit an output following the pre-determined z.
- **In both training and inference:** Optimize the decoding cross-entropy for the chosen context:
  From:
  $$\mathcal{L} = -\sum_t \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x})$$
  To:
  $$\mathcal{L} = -\sum_t \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}_{\leq \boldsymbol{z}_t})$$

- **In both training and inference:** Optimize the decoding cross-entropy for the chosen context:

$$\mathcal{L} = -\sum_{t} \log p(y_t | \boldsymbol{y}_{<t}, \boldsymbol{x}_{\leq z_t}) \qquad \text{(Path)}$$

We find it beneficial to include larger contexts in the optimization:

$$\mathcal{L} = -\sum_{t} \sum_{c \geq z_t} \log p(y_t | \boldsymbol{y}_{<t}, \boldsymbol{x}_{\leq c}) \qquad \text{(Above path)}$$



Dynamic z

Build an HMM with latent variables $\boldsymbol{z}$ and observations $\boldsymbol{y}$.

1. At every position (t, j) make a decision read=1/write=0:

$$\rho_{tj} = \sigma(C^\top h_{tj}),$$

2. Estimate the HMM transition probabilities from read/write decisions:

$$M_{tjk} = p_\theta(z_{t+1} = k | z_t = j, \boldsymbol{y}_{<t}, \boldsymbol{x}) = \begin{cases} (1 - \rho_{tk}) \prod_{l=j}^{k-1} \rho_{tl}, & \text{if } k \geq j, \\ 0, & \text{otherwise (non-decreasing } \boldsymbol{z}) \end{cases}$$

Dynamic z

Build an HMM with latent variables z and observations y.

3. EM algorithm:

$$\begin{aligned}\log p_\theta(\boldsymbol{y}|\boldsymbol{x}) &\geq \log p(\boldsymbol{y}|\boldsymbol{x}) - \mathcal{D}_{\mathrm{KL}}\left(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})\right) \\ &\geq \mathcal{H}(q) + \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x})\end{aligned}$$

Dynamic z

Build an HMM with latent variables z and observations y.

3. EM algorithm:

$$\begin{aligned}\log p_\theta(\boldsymbol{y}|\boldsymbol{x}) &\geq \log p(\boldsymbol{y}|\boldsymbol{x}) - \overbrace{\mathcal{D}_{\mathrm{KL}}\left(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})\right)}^{0} \\ &\geq \mathcal{H}(q) + \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x})\end{aligned}$$

1. E-step: $q := p_\theta(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})$. Infer the posterior via Baum-Welch.

Dynamic z

Build an HMM with latent variables $\boldsymbol{z}$ and observations $\boldsymbol{y}$.

3. EM algorithm:

$$\begin{aligned}\log p_\theta(\boldsymbol{y}|\boldsymbol{x}) &\geq \log p(\boldsymbol{y}|\boldsymbol{x}) - \overbrace{\mathcal{D}_{\mathrm{KL}}\left(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})\right)}^{0} \\ &\geq \mathcal{H}(q) + \underbrace{\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x})}_{Q}\end{aligned}$$

1. E-step: $q := p_\theta(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})$. Infer the posterior via Baum-Welch.
2. M-step: $\theta := \arg\max_\theta Q(\theta)$. SGD

$$Q(q, \theta) = \mathcal{W}(q, \theta) + \mathcal{C}(q, \theta).$$

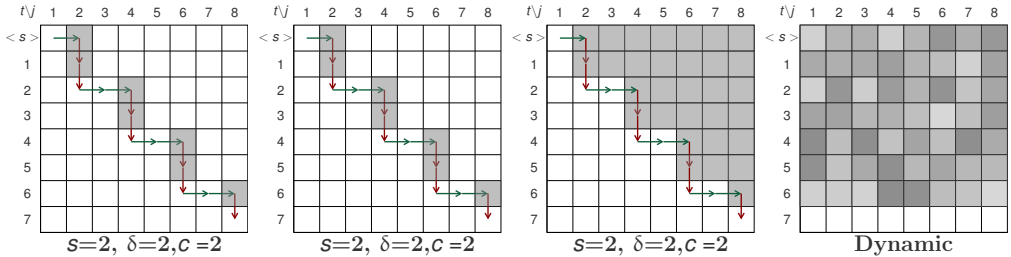
$$\mathcal{W}(q, \theta) = -\sum_{t=1}^{|\boldsymbol{y}|} \sum_{j=1}^{|\boldsymbol{x}|} q(z_t = j) \log p_\theta(y_t | \boldsymbol{y}_{<t}, \boldsymbol{x}_{\leq j}). \qquad \text{(writing)}$$

$$\mathcal{C}(q, \phi) = \sum_{t=1}^{|\boldsymbol{y}|-1} \sum_{j=1}^{|\boldsymbol{x}|} A_{tj} \log \rho_{tj} + B_{tj} \log(1 - \rho_{tj}) \qquad \text{(controlling)}$$

$$A_{tj} = \sum_{l \leq j} \sum_{k > j} q(z_t{=}l, z_{t+1}{=}k) \qquad ((t, j) \rightarrow (t, j+1))$$

$$B_{tj} = q(z_{t+1} = j) \qquad ((t, j) \downarrow (t+1, j))$$

The controller | recap



(a) Masked pervasive attention (MPA)

(b) Deterministic - Path

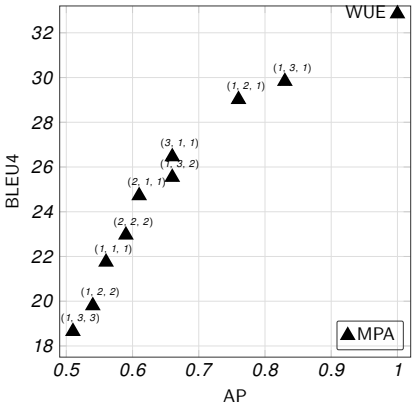
(c) Deterministic - Above path

(d) Dynamic

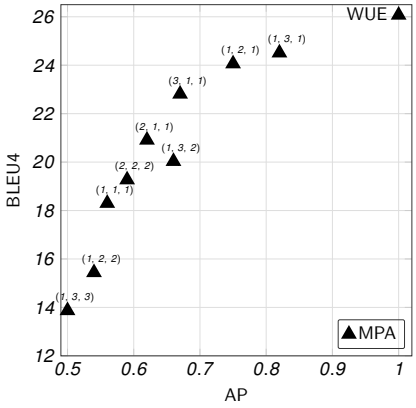
Dataset	Vocab (BPE)	Train	Valid	Test
IWSLT14 De-En/En-De	En(6632), De(8848)	160239	7283	6633
WMT15 Ru-En/En-ru	En(18480), Ru(20000)	806996	8182	3000

- For training, we filter out $(\boldsymbol{x}, \boldsymbol{y})$ if $|\boldsymbol{x}| > 50$ or $|\boldsymbol{y}| > 50$.
- Inference with a beam of size 5
- Evaluate the translation quality with case-sensitive tokenzied BLEU.
- Decoding latency measured with the average proportion (AP):

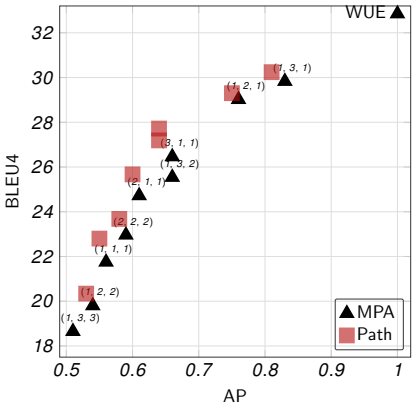
$$AP = \frac{1}{|\boldsymbol{y}|} \sum_{t}^{|\boldsymbol{y}|} z_t / |\boldsymbol{x}|.$$



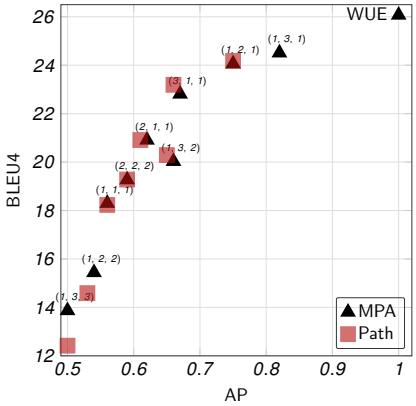
(a) IWSLT14 De-En



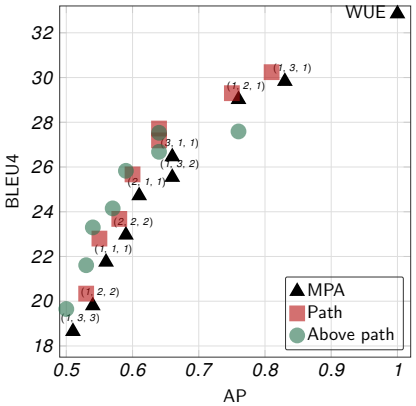
(b) IWSLT14 En-De



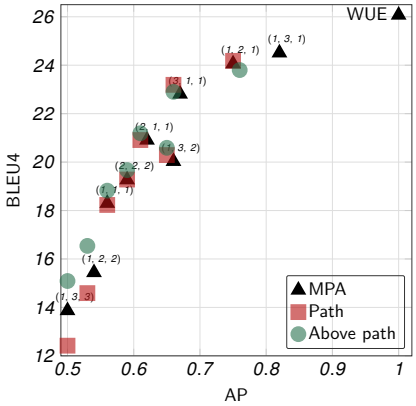
(a) IWSLT14 De-En



(b) IWSLT14 En-De

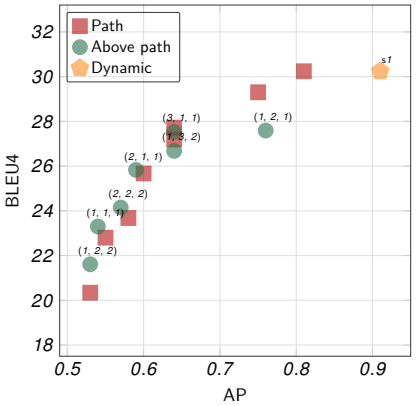


(a) IWSLT14 De-En

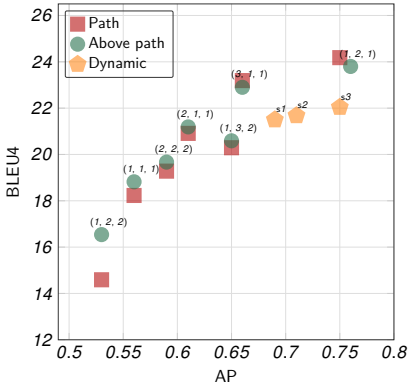


(b) IWSLT14 En-De

Experiments | IWSLT14 De-En/En-De



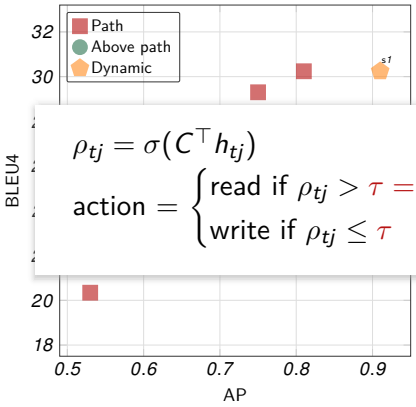
(a) IWSLT14 De-En



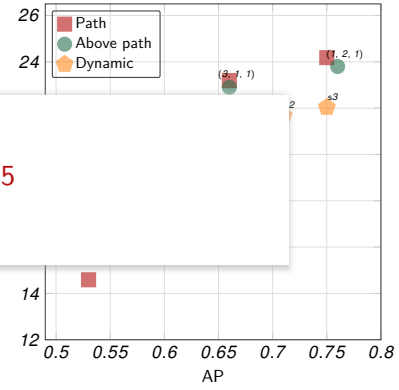
(b) IWSLT14 En-De

$$\rho_{tj} = \sigma(C^\top h_{tj})$$

$$\text{action} = \begin{cases} \text{read if } \rho_{tj} > \tau = 0.5 \\ \text{write if } \rho_{tj} \leq \tau \end{cases}$$

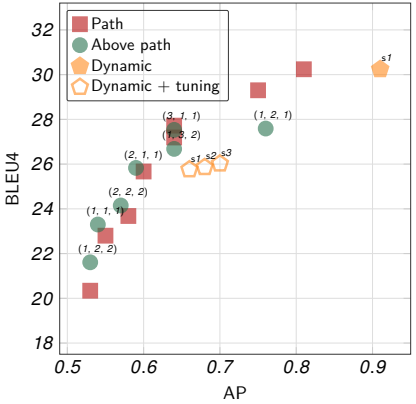


(a) IWSLT14 De-En

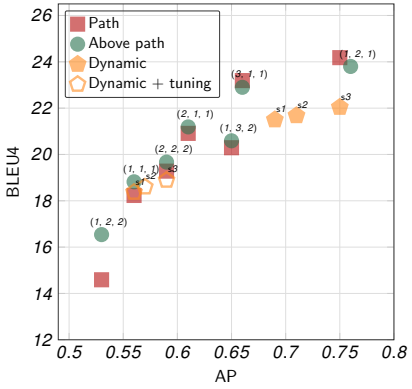


(b) IWSLT14 En-De

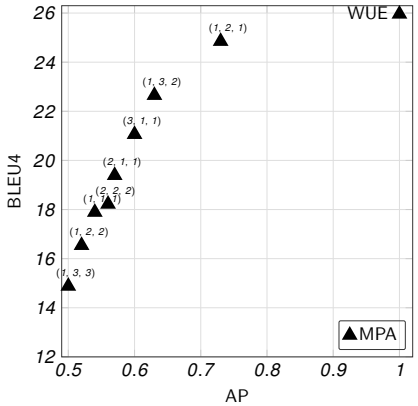
Experiments | IWSLT14 De-En/En-De



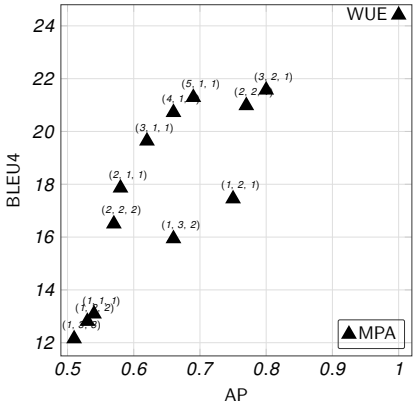
(a) IWSLT14 De-En



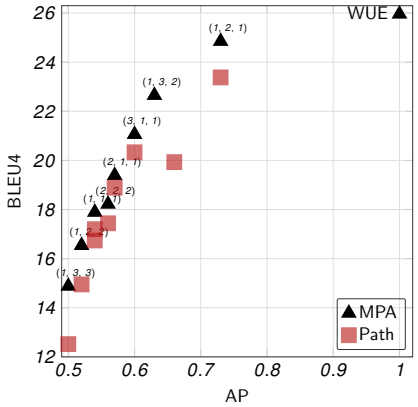
(b) IWSLT14 En-De



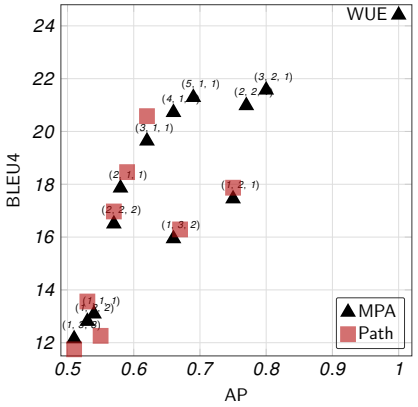
(a) WMT15 Ru-En



(b) WMT15 En-Ru

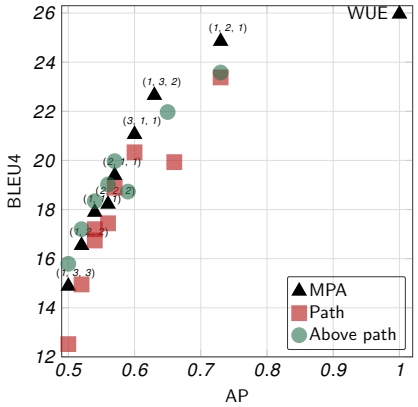


(a) WMT15 Ru-En

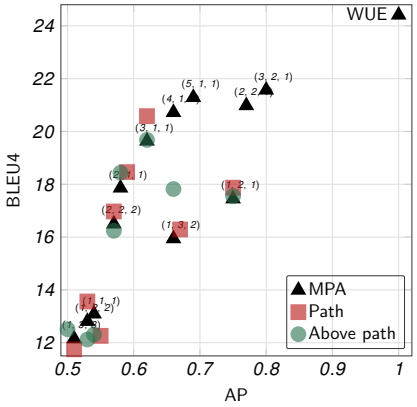


(b) WMT15 En-Ru

Experiments | WMT15 Ru-En/En-Ru

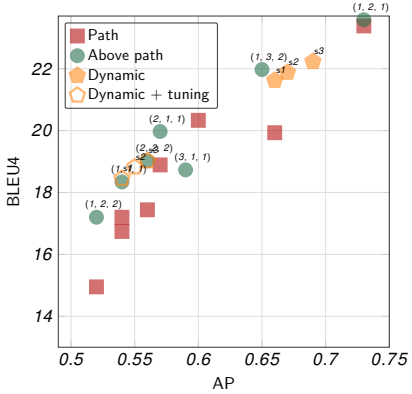


(a) WMT15 Ru-En

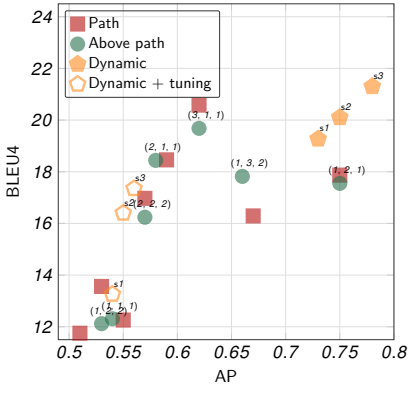


(b) WMT15 En-Ru

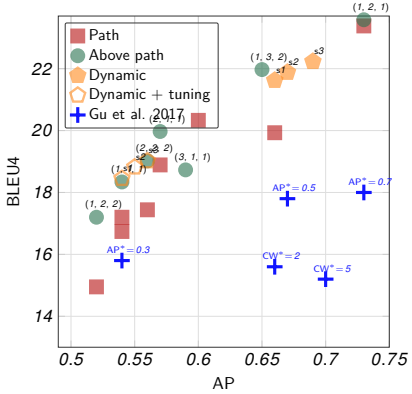
Experiments | WMT15 Ru-En/En-Ru



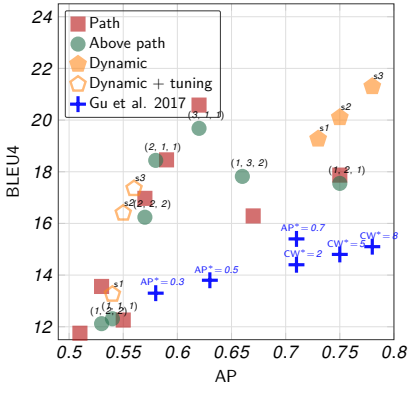
(a) WMT15 Ru-En



(b) WMT15 En-Ru

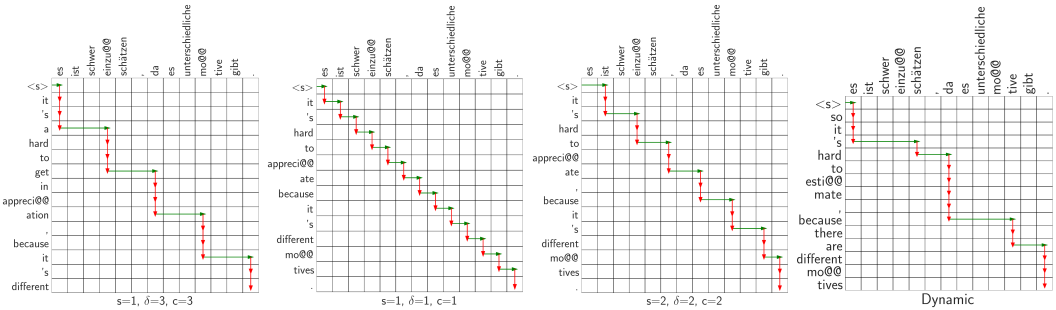


(a) WMT15 Ru-En



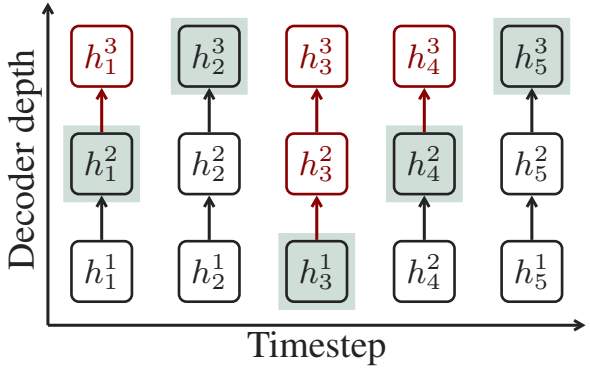
(b) WMT15 En-Ru

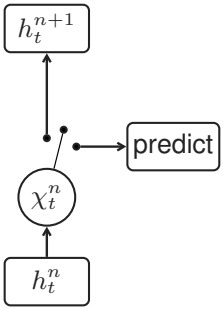
Conclusion



- Both deterministic and dynamic models achieve a good BLEU-AP trade-off i.e correct translations with low delay.
- Future work: Analyze whether read/write decisions reveal any meaningful linguistic structure.

facebook Artificial Intelligence





Thank you for your attention.

Questions / Feedback.