Depth-Adaptive Transformer

For resource efficient machine translation

facebook
Artificial Intelligence Research

Maha Elbayad*   Jiatao Gu   Michael Auli

Submitted to ICLR 2020

October 4, 2019

* work done during an internship at FAIR, Menlo Park, CA

Motivation

State-of-the-art seq2seq models process *easy* and *hard* samples the same way.

Easy: Merci. $\rightarrow$ Thank you.
Hard: Il s'agit là de rien de moins que de réinventer l'Union européenne sans détruire celle qui existe déjà! $\rightarrow$ In doing so, we need, no less, to reinvent the European Union, but without destroying the present Union.

Examples from WMT14 En-FR

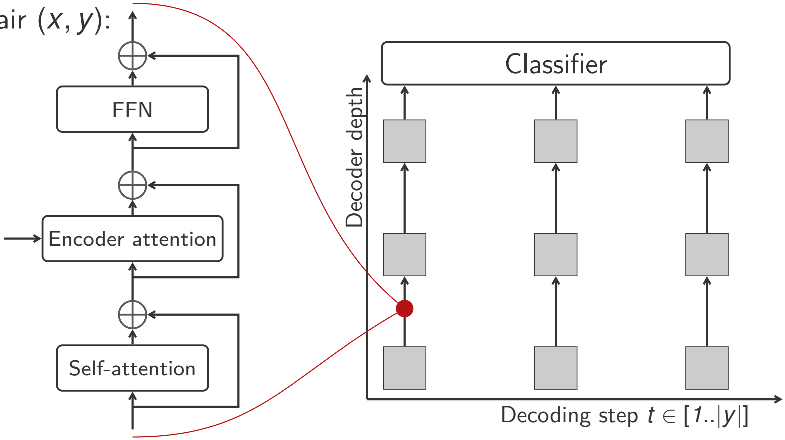
Our goals:

1. Train a seq2seq model capable of yielding an output at varying levels of computation.
2. Plug a module on top of the seq2seq model to choose the 'appropriate' amount of computation.

Slicing the Transformer decoder

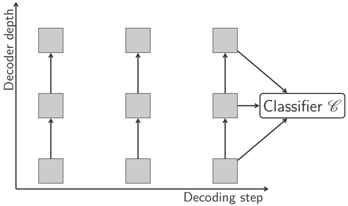
For a source-target pair (x, y):

source hidden states
$\{s_1, s_2, \ldots, s_{|x|}\}$

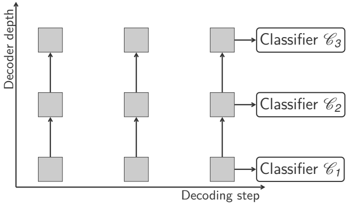


Slicing the Transformer decoder

1. Train a seq2seq model capable of yielding an output at varying levels of computation.



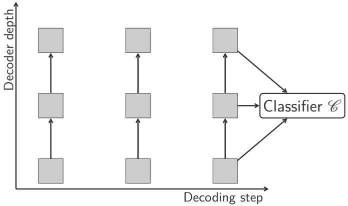
With a single classifier $\mathscr{C}$



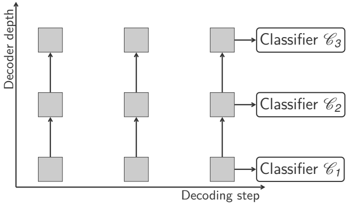
Inserting intermediate classifiers $\mathscr{C}_1, \mathscr{C}_2$

Slicing the Transformer decoder

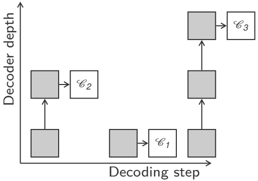
1. Train a seq2seq model capable of yielding an output at varying levels of computation.



With a single classifier $\mathscr{C}$

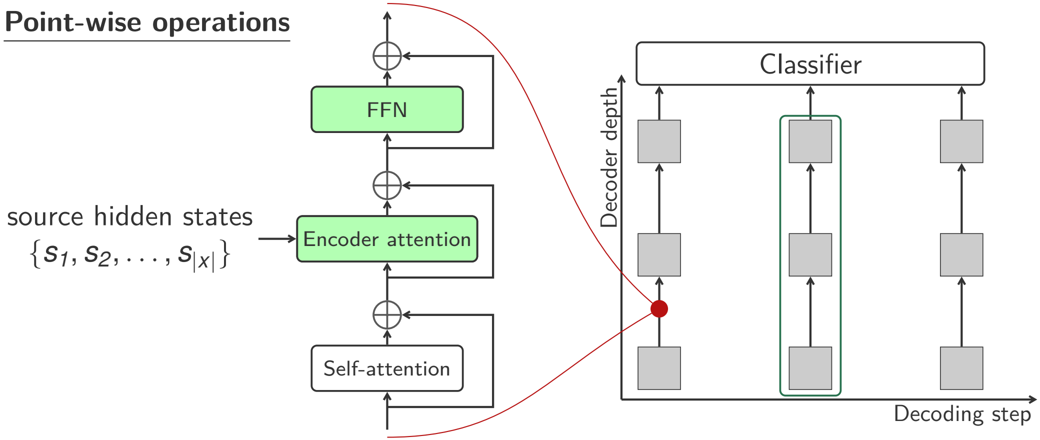


Inserting intermediate classifiers $\mathscr{C}_1, \mathscr{C}_2$



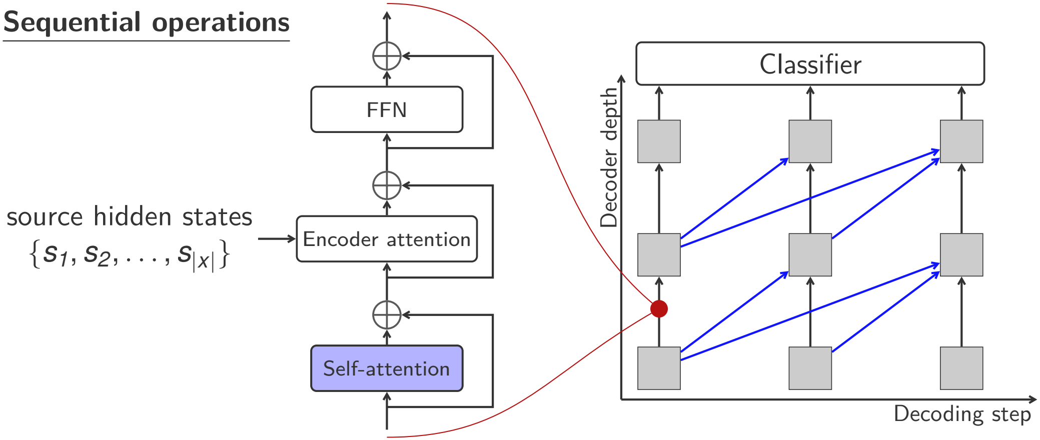
Final outcome

Slicing the Transformer decoder



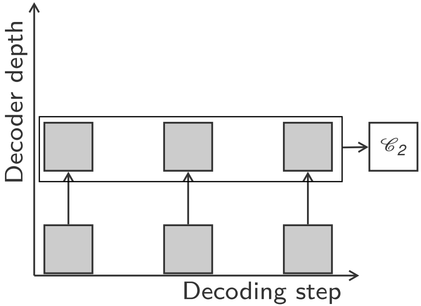
Slicing the Transformer decoder

Sequential operations



Slicing the Transformer decoder

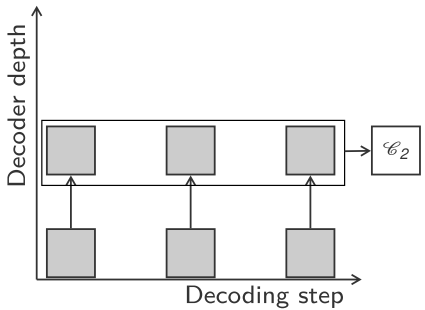
Issue: How to address the interaction between steps in self-attention?



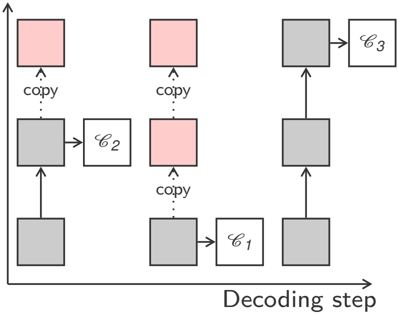
Make a **sequence-specific decision**
All tokens exit at the same point

Slicing the Transformer decoder

Issue: How to address the interaction between steps in self-attention?

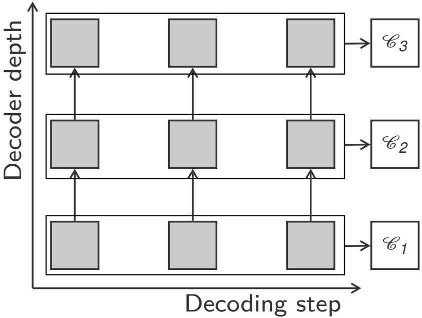


Make a **sequence-specific decision**
All tokens exit at the same point



Make **token-specific decisions**
Copy activations to the subsequent blocks.

Let h_t^n be the hidden state at time-step t (encoding the source $\boldsymbol{x}$ and the prefix $\boldsymbol{y}_{\leq t}$) after going through n blocks (out of N).

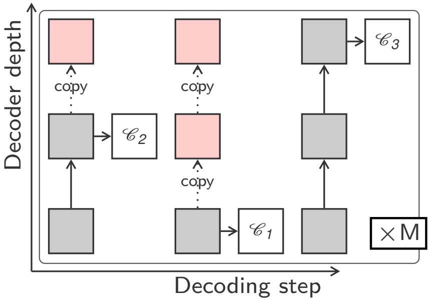


For $n \in 1 \dots N$:

1. $\mathrm{LL}^n = \sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t | h_{t-1}^n)$

$$\mathcal{L}_{dec}(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{n=1}^{N} \omega_n \, \mathrm{LL}^n,$$

where $\{\omega_n\}_n$ weigh the different losses.

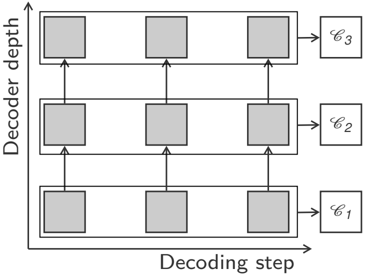


For $m \in 1 \ldots M$:

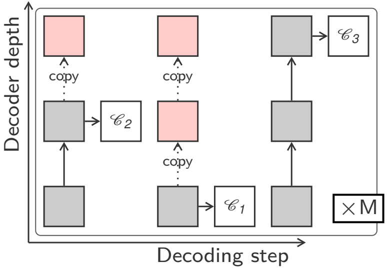
1. Sample a sequence of exits $(n_1, n_2, \ldots, n_{|\mathbf{y}|}) \sim \mathcal{U}([1..N])^{|\mathbf{y}|}$
2. $\mathrm{LL}^{(m)} = \sum_{t=1}^{|\mathbf{y}|} \log p(y_t | h_{t-1}^{n_t})$

$$\mathcal{L}_{dec}(\mathbf{x}, \mathbf{y}) = -\frac{1}{M} \sum_{m=1}^{M} \mathrm{LL}^{(m)}$$

Pre-training the anytime decoder | Mixed vs. Aligned training

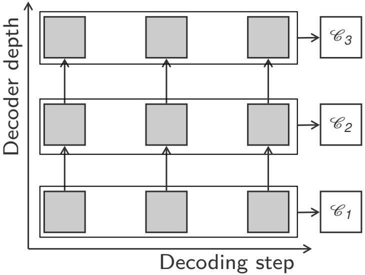


- Adapted for sequence-specific decoding.
- A single forward pass.

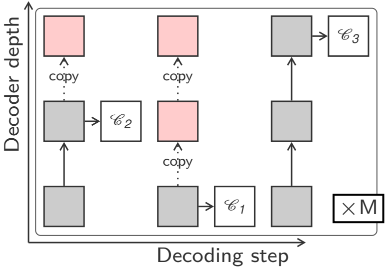


- Adapted for token-specific decoding.
- Requires multiple forward passes.

Pre-training the anytime decoder | Mixed vs. Aligned training



- Adapted for sequence-specific decoding.
- A single forward pass.



- Adapted for token-specific decoding.
- Requires multiple forward passes.

Better performances with the 'aligned' training.

Exit prediction

1. Train a seq2seq model capable of yielding an output at varying levels of computation.
2. Plug a module on top of the seq2seq model to choose the 'appropriate' amount of computation.

We present 3 approaches with oracle-supervised **trainable classifiers** and 1 approach based on **confidence thresholding**.

Predict a single exit for all tokens in a given sample.

1. Model the exit distribution q : predict the exit given an aggregate of the source hidden states $\{s_1, \ldots, s_{|\boldsymbol{x}|}\}$:

$$s = \frac{1}{|\boldsymbol{x}|} \sum_{i=1}^{|\boldsymbol{x}|} s_i, \quad q(n|\boldsymbol{x}) = \mathrm{softmax}(W_h s + b_h),$$

with weights and biases (W_h, b_h) such that W_h maps to $\mathbb{R}^N$.

Trainable exit prediction | Sequence-specific depth

Predict a single exit for all tokens in a given sample.

1. Model the exit distribution q : predict the exit given an aggregate of the source hidden states $\{s_1, \ldots, s_{|\boldsymbol{x}|}\}$:
2. Evaluate a target distribution q^* (oracle-based):

Predict a single exit for all tokens in a given sample.

1. Model the exit distribution q : predict the exit given an aggregate of the source hidden states $\{s_1, \ldots, s_{|\boldsymbol{x}|}\}$:
2. Evaluate a target distribution q^* (oracle-based):

Likelihood: $\mathrm{LL}^n = \sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t | h_{t-1}^n)$, $\quad q^*(\boldsymbol{x}, \boldsymbol{y}) = \delta(\arg\max_n \mathrm{LL}^n - \lambda n)$

Predict a single exit for all tokens in a given sample.

1. Model the exit distribution q : predict the exit given an aggregate of the source hidden states $\{s_1, \ldots, s_{|\boldsymbol{x}|}\}$:
2. Evaluate a target distribution q^* (oracle-based):

$$\text{Likelihood:} \quad \mathrm{LL}^n = \sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t | h_{t-1}^n), \qquad q^*(\boldsymbol{x}, \boldsymbol{y}) = \delta(\arg\max_n \mathrm{LL}^n - \lambda n)$$

$$\text{Correctness:} \quad C^n = |\{t \mid y_t = \arg\max_y p(y | h_{t-1}^n)\}|, \quad q^*(\boldsymbol{x}, \boldsymbol{y}) = \delta(\arg\max_n C^n - \lambda n)$$

Predict a single exit for all tokens in a given sample.

1. Model the exit distribution q : predict the exit given an aggregate of the source hidden states $\{s_1, \dots, s_{|\boldsymbol{x}|}\}$:
2. Evaluate a target distribution q^* (oracle-based):

$$\text{Likelihood:} \quad \mathrm{LL}^n = \sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t|h_{t-1}^n), \qquad q^*(\boldsymbol{x},\boldsymbol{y}) = \delta(\arg\max_n \mathrm{LL}^n - \lambda n)$$

$$\text{Correctness:} \quad C^n = |\{t \mid y_t = \arg\max_y p(y|h_{t-1}^n)\}|, \quad q^*(\boldsymbol{x},\boldsymbol{y}) = \delta(\arg\max_n C^n - \lambda n)$$

3. Optimize $H(q^*, q)$.

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$:
   With a multinomial: prediction after the 1st block.
   $q_t(n|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \text{softmax}(W_h h_t^1 + b_h)$ with h_t^1 the output of the first decoder block.

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$:
   With a multinomial: prediction after the 1st block.
   $q_t(n|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \text{softmax}(W_h h_t^1 + b_h)$ with h_t^1 the output of the first decoder block.

   With a Poisson binomial (+ monotonicity constraints):
   Estimate a halting probability after each block:

$$\forall n \in [1..N{-}1],\ \chi_t^n = \sigma(W_h h_t^n + b_h)$$

$$q_t(n|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \begin{cases} \chi_t^n \prod\limits_{n'<n} (1 - \chi_t^{n'}), & \text{if } n < N \\ \prod\limits_{n'<N} (1 - \chi_t^{n'}), & \text{if } n = N \end{cases}$$

Trainable exit prediction | Token-specific depth

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$.
2. Evaluate a target distribution $q_t^*, \forall t$ (oracle-based):

$$\text{Likelihood:} \quad \text{LL}_t^n = \log p(y_t | h_t^{n-1})$$

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$.
2. Evaluate a target distribution $q_t^*, \forall t$ (oracle-based):

$$\text{Likelihood:} \quad \mathrm{LL}_t^n = \log p(y_t | h_t^{n-1})$$

$$\text{Smoothed likelihood:} \quad \mathrm{smoothLL}_t^n = \sum_{t'} \kappa(t, t')\, \mathrm{LL}_{t'}^n, \; \kappa(t, t') = e^{-\frac{|t-t'|^2}{\sigma}}$$

Trainable exit prediction | Token-specific depth

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$.
2. Evaluate a target distribution $q_t^*, \forall t$ (oracle-based):

$$\text{Likelihood:} \quad \mathrm{LL}_t^n = \log p(y_t | h_t^{n-1})$$

$$\text{Smoothed likelihood:} \quad \mathrm{smoothLL}_t^n = \textstyle\sum_{t'} \kappa(t, t')\, \mathrm{LL}_{t'}^n, \; \kappa(t, t') = e^{-\frac{|t-t'|^2}{\sigma}}$$

$$\text{Correctness:} \quad C_t^n = (y_t = \arg\max_y p(y | h_{t-1}^n))$$

Trainable exit prediction | Token-specific depth

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$.
2. Evaluate a target distribution $q_t^*, \forall t$ (oracle-based):

$$\begin{aligned}
\text{Likelihood:}\quad & \mathrm{LL}_t^n = \log p(y_t | h_t^{n-1}) \\
\text{Smoothed likelihood:}\quad & \mathrm{smoothLL}_t^n = \textstyle\sum_{t'} \kappa(t,t')\, \mathrm{LL}_{t'}^n,\ \kappa(t,t') = e^{-\frac{|t-t'|^2}{\sigma}} \\
\text{Correctness:}\quad & C_t^n = (y_t = \arg\max_y p(y | h_{t-1}^n)) \\
\text{Smoothed correctness:}\quad & \mathrm{smoothC}_t^n = \textstyle\sum_{t'} \kappa(t,t')\ C_t^n
\end{aligned}$$

Predict an exit for each token.

1. Model the exit distribution $q_t, \forall t$.
2. Evaluate a target distribution $q_t^*, \forall t$ (oracle-based):

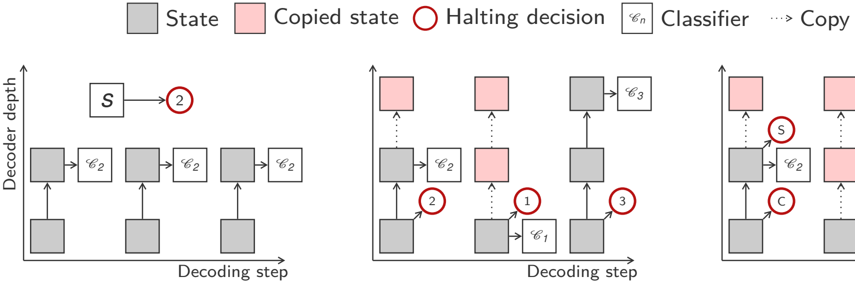
$$\begin{aligned}
\text{Likelihood:} &\quad \mathrm{LL}_t^n = \log p(y_t | h_t^{n-1}) \\
\text{Smoothed likelihood:} &\quad \mathrm{smoothLL}_t^n = \textstyle\sum_{t'} \kappa(t, t')\, \mathrm{LL}_{t'}^n,\ \kappa(t, t') = e^{-\frac{|t-t'|^2}{\sigma}} \\
\text{Correctness:} &\quad C_t^n = (y_t = \arg\max_y p(y | h_{t-1}^n)) \\
\text{Smoothed correctness:} &\quad \mathrm{smooth}C_t^n = \textstyle\sum_{t'} \kappa(t, t')\ C_t^n
\end{aligned}$$

▷ Regularize the score and estimate a target distribution:

$$q_t^*(\boldsymbol{x}, \boldsymbol{y}) = \delta(\arg\max_n \mathrm{score}_t^n - \lambda n)$$

3. Optimize $\sum_t H(q_t^*, q_t)$.

Adaptive exit prediction | inference



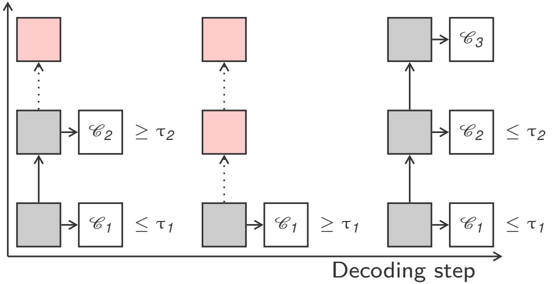
(a) Sequence-specific depth

(b) Token-specific - Multinomial $\sigma \in [2, 3]$

(c) Token-specific - Poisson $\sigma \in [2, 3]$

Confidence thresholding

Input: $\boldsymbol{x}, \tau = (\tau_1, \tau_2, \ldots, \tau_{N-1})$
for $n \in 1 \ldots N$ **do**
  Forward through the nth block.
  if $\arg\max_{y_t} p(y_t | h^n_{t-1}) \geq \tau_n$ **then**
    Exit
  end if
end for



Tune τ with random search on the development set so as to maximize BLEU.

An extension of the thresholding in 'Multi-Scale Dense Networks for Resource Efficient Image Classification' (Huang et al. ICLR'18) to sequence prediction.

Experiments on IWSLT14 DE$\rightarrow$ EN

- Train (160K), Dev (7K), Test (6K)
- Vocabularies: Joint byte-pair encoding: EN 8K & DE 6.7K
- Average sequence length 23 tokens
- Architecture: Transformer *small*
$$N = 6,\ d_{\text{enc}} = 512,\ d_{\text{dec}} = 256,\ d_{\text{ffn}} = 1024.$$
- Separate 6 anytime classifiers $\mathscr{C}_1, \ldots, \mathscr{C}_6$.
- Evaluation: Best checkpoint on dev with beam=5.

- N baselines : N independent models with varying depths $n \in [1 \ldots N]$

Pre-training the anytime decoder | Mixed vs. Aligned training | IWSLT14

For each aligned/mixed model:

- Uniform: evaluating with a random exit per token $n_t \sim \mathcal{U}([1 \dots N])$.
- $n=$: evaluating each exit independently.

	Uniform	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	Average
Baseline	-	34.5	35.5	35.8	35.7	35.8	36.0	35.5
Aligned ($\omega_n = 1/N$)	**35.5**	34.1	**35.5**	**35.8**	**36.1**	**36.1**	**36.2**	**35.6**

BLEU on the development set of IWSLT14 De-En

For each aligned/mixed model:

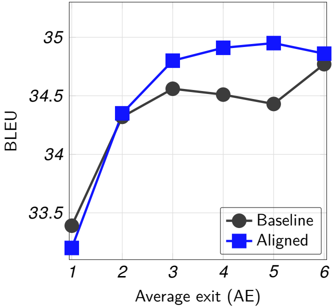
- Uniform: evaluating with a random exit per token $n_t \sim \mathcal{U}([1 \ldots N])$.
- $n=$: evaluating each exit independently.

	Uniform	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	Average
Baseline	-	34.5	35.5	35.8	35.7	35.8	36.0	35.5
Aligned ($\omega_n = 1/N$)	**35.5**	34.1	**35.5**	**35.8**	**36.1**	**36.1**	**36.2**	**35.6**
Mixed $M=1$	34.1	32.9	34.3	34.5	34.5	34.6	34.5	34.2
Mixed $M=3$	35.1	33.9	35.2	35.4	35.5	35.5	35.5	35.2
Mixed $M=6$	35.3	**34.2**	35.4	**35.8**	35.9	35.8	35.9	35.5
Mixed $M=8$	35.2	33.9	35.1	35.4	35.6	35.7	35.7	35.2

BLEU on the development set of IWSLT14 De-En

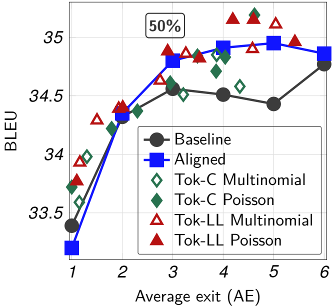
Training for adaptive exits | IWSLT14

- Finetuning an aligned model with $\mathcal{L} = \mathcal{L}_{\text{dec}} + H(q^*, q)$
- Measuring translation quality with BLEU (the higher the better) and the computational cost with the average exit AE (the lower the better).



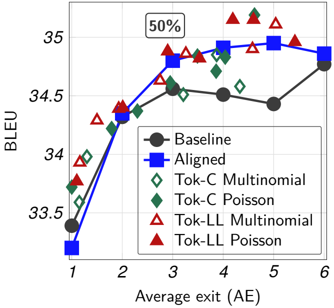
(a) Token-specific (test)

- Finetuning an aligned model with $\mathcal{L} = \mathcal{L}_{\text{dec}} + H(q^*, q)$
- Measuring translation quality with BLEU (the higher the better) and the computational cost with the average exit AE (the lower the better).

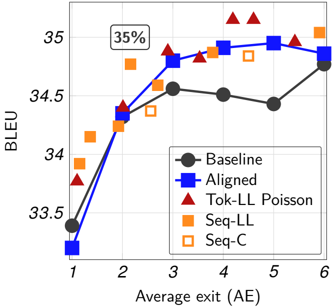


(a) Token-specific (test)

- Finetuning an aligned model with $\mathcal{L} = \mathcal{L}_{\text{dec}} + H(q^*, q)$
- Measuring translation quality with BLEU (the higher the better) and the computational cost with the average exit AE (the lower the better).



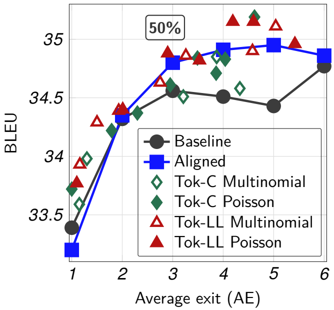
(a) Token-specific (test)



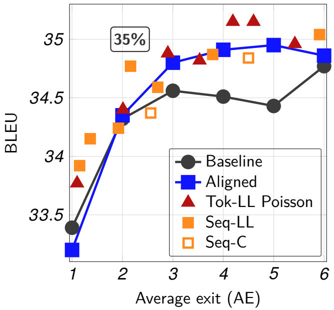
(b) Sequence-specific depth (test)

Training for adaptive exits | IWSLT14

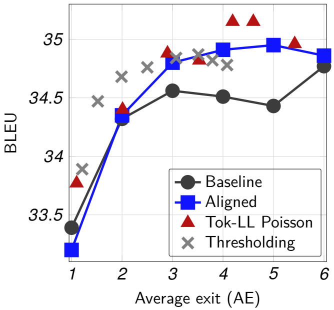
- Finetuning an aligned model with $\mathcal{L} = \mathcal{L}_{\text{dec}} + H(q^*, q)$
- Measuring translation quality with BLEU (the higher the better) and the computational cost with the average exit AE (the lower the better).



(a) Token-specific (test)



(b) Sequence-specific depth (test)



(c) Confidence thresholding (test)

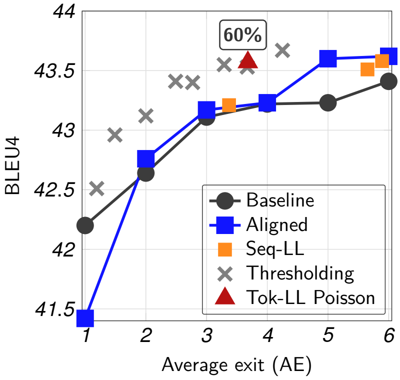
- Train (35.5M), Dev (26K), Test (newstest14: 3K)
- Vocabularies: Joint byte-pair encoding + shared dictionary: 44K
- Average sequence length 29 tokens
- Architecture: Transformer *big*

$$N = 6,\ d_{\text{enc}} = 1024 = d_{\text{dec}} = 1024,\ d_{\text{ffn}} = 4096.$$

- Tied anytime classifiers $\mathscr{C}_1 = \mathscr{C}_2 = \ldots \mathscr{C}_6$.
- Evaluation: average of 10 checkpoints with beam=4 and length-penalty=0.6.

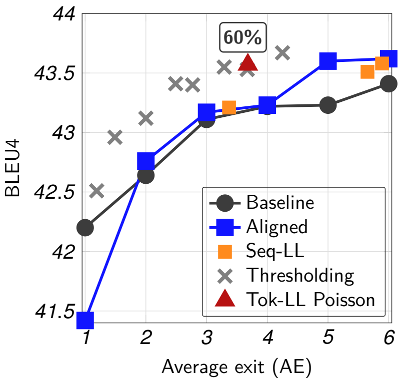
- N baselines : N independent models with varying depths $n \in [1 \ldots N]$

Experiments on WMT14 EN→ FR | Scaling depth-adaptive models

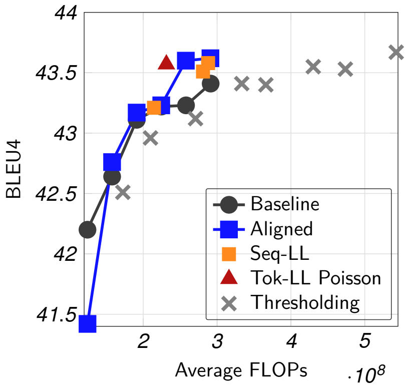


(a) BLEU vs. AE (test)

Experiments on WMT14 EN$\rightarrow$ FR | Scaling depth-adaptive models

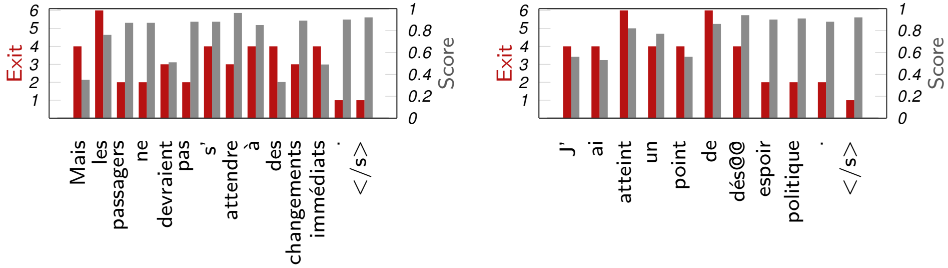


(a) BLEU vs. AE (test)



(b) BLEU vs. FLOPs (test)

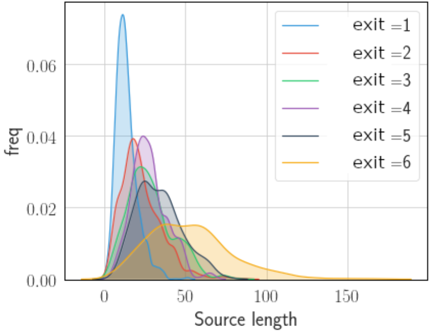
Qualitative results | Token-specific exit



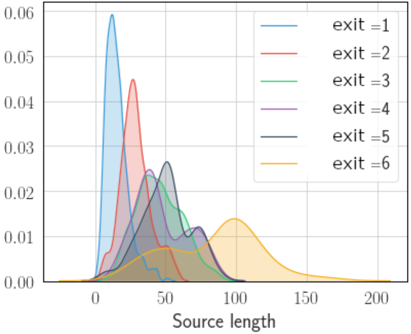
src: But passengers shoul@@dn't expect changes to happen immediately.
ref: Mais les passagers ne devraient pas s' attendre à des changements immédiats.

src: I 've reached a point of political des@@ pair.
ref: Je suis au bord du dés@@ espoir politique.

Figure: Examples from the WMT14 En-Fr with Tok-LL Poisson



(a) Seq-LL ($\lambda = 2$)



(b) Seq-LL ($\lambda = 1$)

Figure: Distribution of the exits wrt. the source sequence length with different regularizers $\lambda = 1$ and $\lambda = 2$. Results on IWSLT14 test set.

Conclusion

- We extended anytime prediction to the structured prediction setting and introduced simple yet effective methods to equip models with the ability to emit outputs at different levels.
- We compared a number of different mechanisms to predict the required network depth and find that a simple likelihood based Poisson classifier obtains the best trade-off between speed and accuracy.
- Our results show that the number of decoder layers can be vastly reduced at no loss in accuracy.