Accelerating Inference in Dynamic Transformer Architectures

Alex Wollam

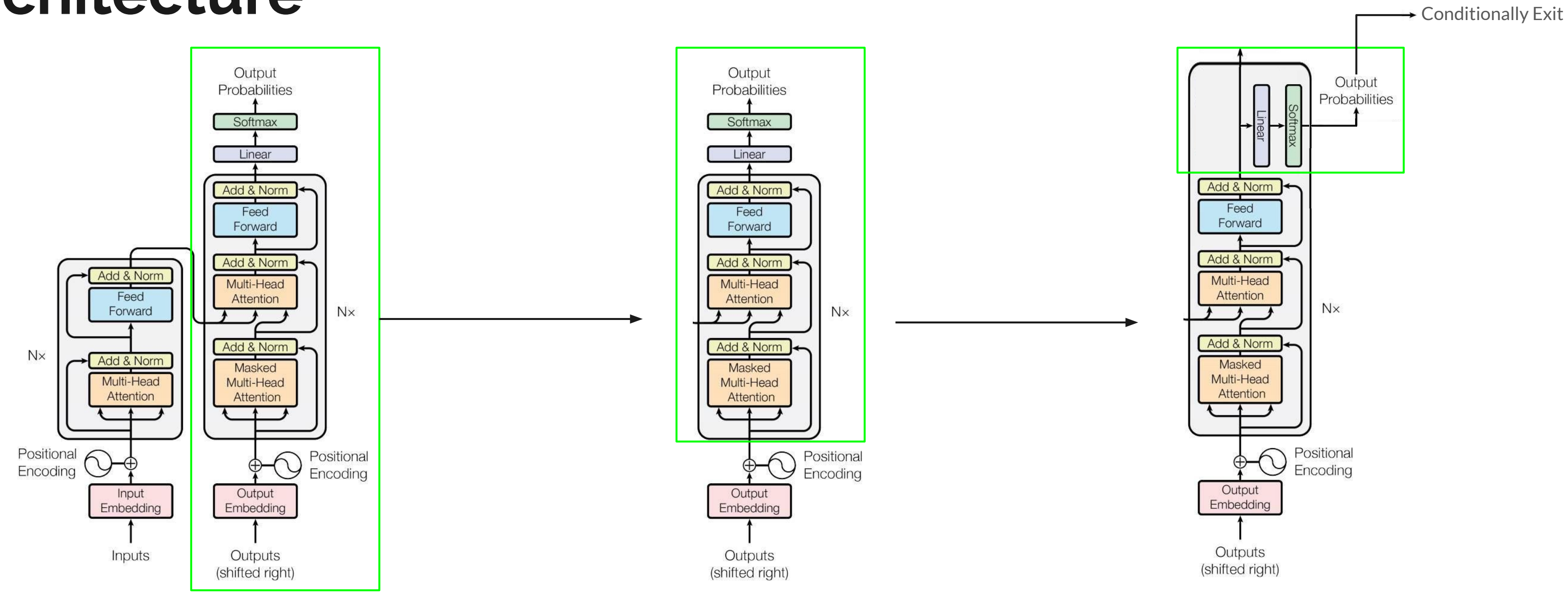
Depth-Adaptive Transformer

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli

Motivation + Contributions

- Modern neural sequence models large + expensive
- Propose transformer-models which adapt the number of layers to each input in order to achieve a good speed-accuracy trade off at inference time

Architecture



Encoder-Decoder Transformer

Decoder

Classify at Each Layer

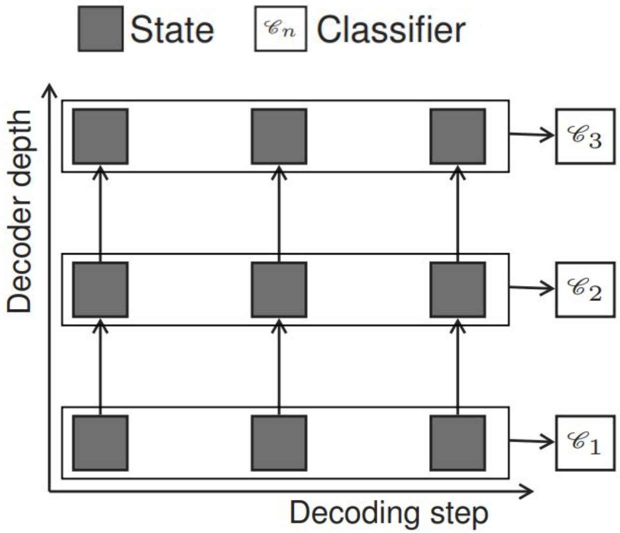
Aligned Training Regime

n := chosen exit, **t** := time-step, **h** := hidden state,
x := source seq, **y** := target seq

$$\mathrm{LL}_t^n = \log p(y_t | h_{t-1}^n),$$

$$\mathrm{LL}^n = \sum_{t=1}^{|\boldsymbol{y}|} \mathrm{LL}_t^n,$$

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



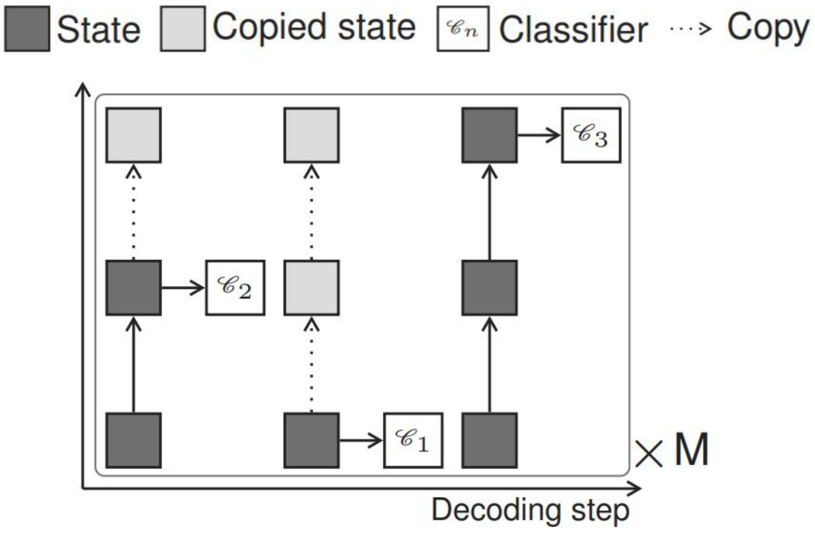
(a) Aligned training

Mixed Training Regime

M := sampled exit sequences

$$\mathrm{LL}(n_1, \ldots, n_{|\boldsymbol{y}|}) = \sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t | h_{t-1}^{n_t}),$$

$$\mathcal{L}_{dec}(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{M} \sum_{m=1}^{M} \mathrm{LL}(n_1^{(m)}, \ldots, n_{|\boldsymbol{y}|}^{(m)}).$$



(b) Mixed training

Adaptive Depth Estimation

Given $q_t(n)$:= the probability of computing *n* blocks before exiting, for token *t* (exit distribution)

$$\mathcal{L}_{\text{exit}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_t H(q_t^*(\boldsymbol{x}, \boldsymbol{y}), q_t(\boldsymbol{x}))$$

H := Cross-entropy

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{L}_{dec}(\boldsymbol{x}, \boldsymbol{y}) + \alpha \mathcal{L}_{\text{exit}}(\boldsymbol{x}, \boldsymbol{y}),$$

Model exit distribution q_t and infer oracle distribution q_t^* via:

- Sequence-specific depth
- Token-specific depth

Sequence-Specific Depth

s := encoder representation/output

Exit distribution q_t:

$$s = \frac{1}{|\boldsymbol{x}|}\sum_t s_t, \quad q(n|\boldsymbol{x}) = \text{softmax}(W_h s + b_h) \in \mathbb{R}^N,$$

weights for halting mechanism

Oracle q_t^*:

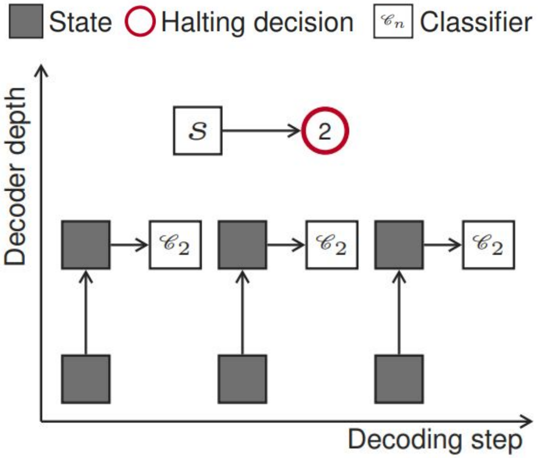
- **Likelihood-based:**

Dirac delta

$$q^*(\boldsymbol{x}, \boldsymbol{y}) = \delta(\arg\max_n \text{LL}^n - \lambda n).$$

- **Correctness-based:**

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



(a) Sequence-specific depth

Token-Specific Depth

Exit distribution q_t:

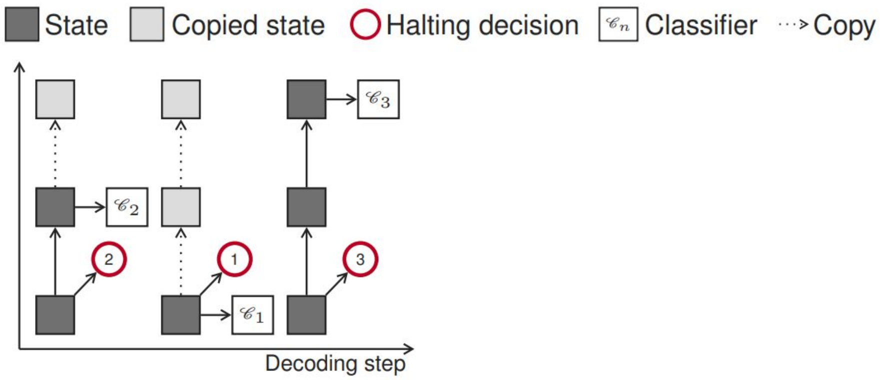
- **Multinomial:**

$$q_t(n|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \text{softmax}(W_h h_t^1 + b_h),$$

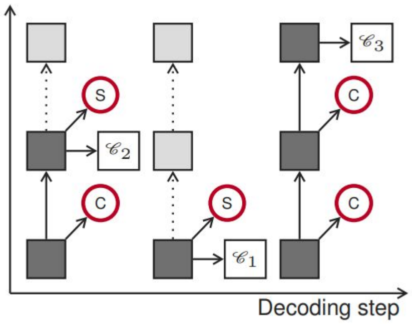
- **Geometric-like:**

$$\forall n \in [1..N-1],\ \chi_t^n = \text{sigmoid}(w_h^\top h_t^n + b_h),$$

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



(b) Token-specific - Multinomial



(c) Token-specific - Geometric-like

Token-Specific Depth

Oracle q_t^*:

- Likelihood-based (LL(σ, λ)):

$$\kappa(t,t') = e^{-\frac{|t-t'|^2}{\sigma}}, \quad \widetilde{\mathrm{LL}}_t^n = \sum_{t'=1}^{|\boldsymbol{y}|} \kappa(t,t')\, \mathrm{LL}_{t'}^n, \quad q_t^*(\boldsymbol{x},\boldsymbol{y}) = \delta(\arg\max_n \widetilde{\mathrm{LL}}_t^n - \lambda n),$$

- Correctness-based:

$$C_t^n = \mathbb{1}[y_t = \arg\max_y p(y|h_{t-1}^n)], \quad \widetilde{C}_t^n = \sum_{t'=1}^{|\boldsymbol{y}|} \kappa(t,t')\, C_t^n,$$

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

Thresholded Depth

- Exit when token output probability exceeds hyper-parameter threshold τ_n

- Thresholds $\boldsymbol{\tau} = (\tau_1, .., \tau_n)$ tuned on valid set to maximize BLEU
  - Sample $\boldsymbol{\tau}$ uniformly across 10K iterations
  - Select that which maximizes performance

Experimental Setup

- Metric:
  - Tokenized BLEU

- Datasets:
  - IWSLT'14 German to English (De-En)
  - WMT'14 English to French (En-Fr)

Experimental Results

	Uniform	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	Average
Baseline	-	34.2	35.3	35.6	35.7	35.6	35.9	35.4
Aligned ($\omega_n = 1$)	**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

IWSLT'14 German to English (De-En)

Experimental Results: Adaptive Depth

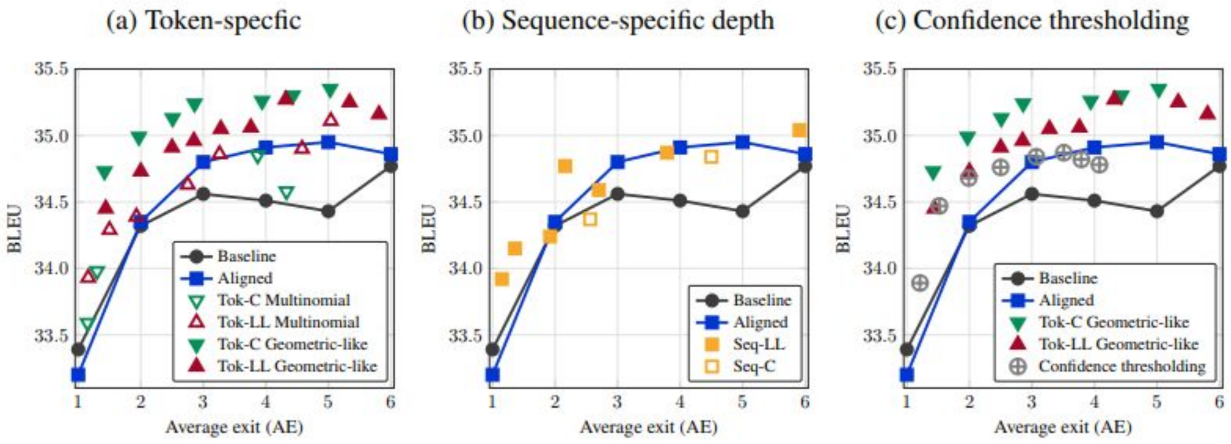


Figure 3: Trade-off between speed (average exit or AE) and accuracy (BLEU) for depth-adaptive methods on the IWSLT14 De-En test set.

Hyperparameter Ablation

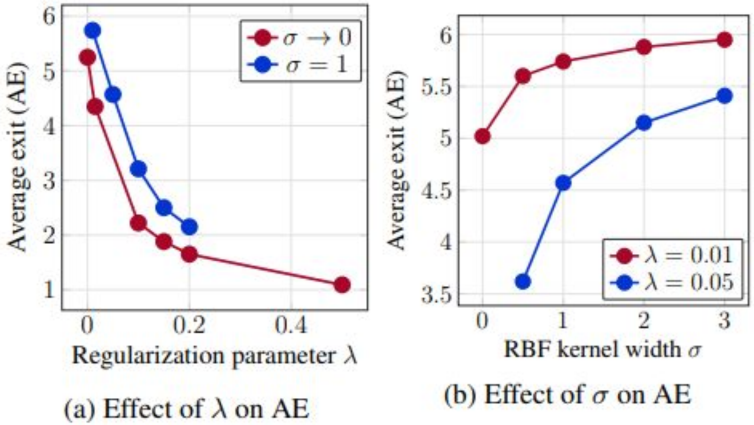


Figure 4: Effect of the hyper-parameters σ and λ on the average exit (AE) measured on the valid set of IWSLT'14 De-En.

Scaling The Depth-Adaptive Models

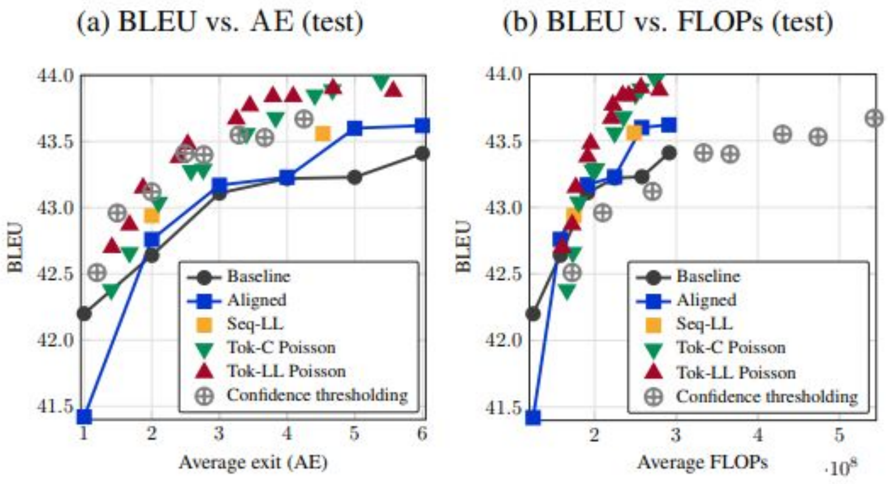
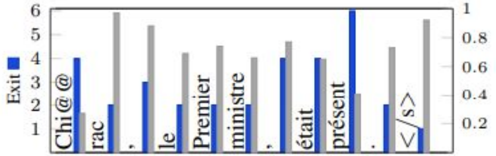
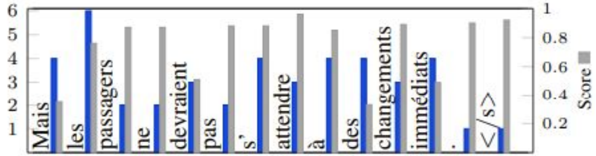


Figure 5: Speed and accuracy on the WMT'14 English-French benchmark (*c.f*. Figure 3).

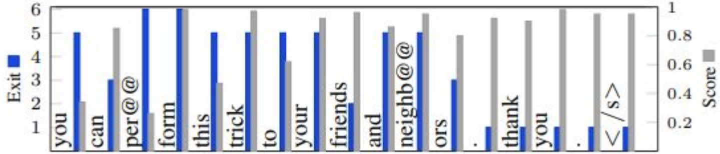
Qualitative Results: Examples



(a) **Src:** Chi@@rac , the Prime Minister , was there .
Ref: Chi@@rac , Premier ministre , est là .

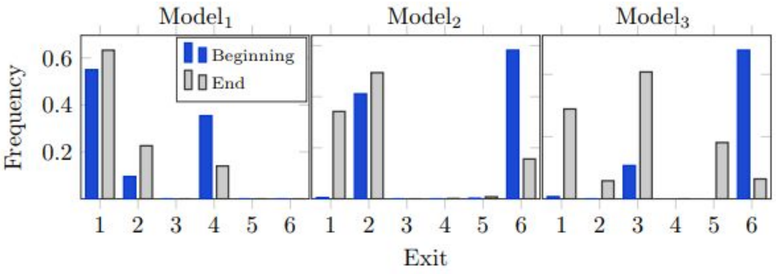


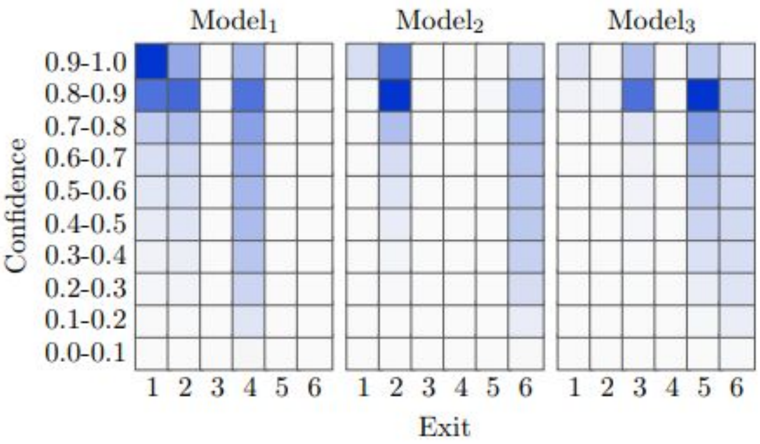
(b) **Src:** But passengers shoul@@dn't expect changes to happen immediately .
Ref: Mais les passagers ne devraient pas s' attendre à des changements immédiats .



Src: diesen trick können sie ihren freunden und nachbarn vor@@führen . danke .
Ref: there is a trick you can do for your friends and neighb@@ors . thanks .

Qualitative Results: Exit Distribution





Conclusion

- Simple methods sufficient for anytime prediction in transformers
- Correctness-based geometric classifier has best speed/accuracy tradeoff
- The number of decoder layers can be reduced by >75% w/out loss in accuracy

DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference

Ji Xin, Raphael Tang , Jaejun Lee , Yaoliang Yu , and Jimmy Lin

Motivation + Contribution

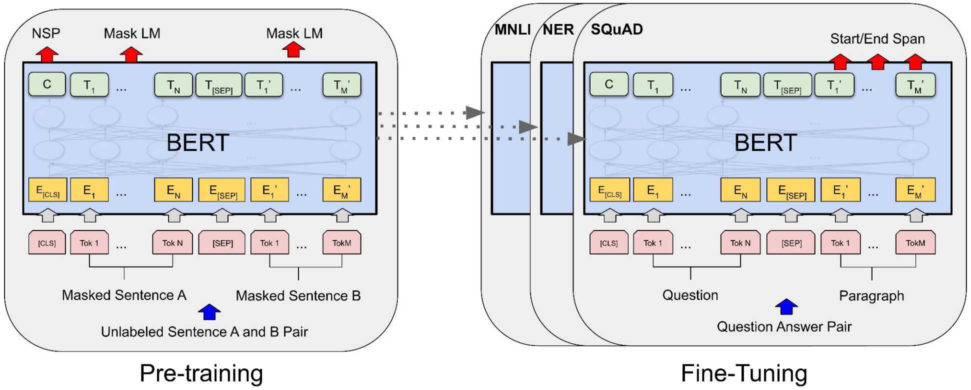
- Large pre-trained models (e.g. BERT) slow in inference

- Accelerate BERT inference w/ early exiting

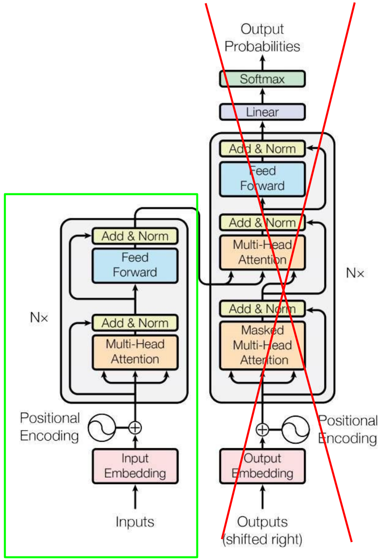
BERT

- Bidirectional Transformer
  - Equivalent to the encoder portion of the original encoder-decoder transformer framework

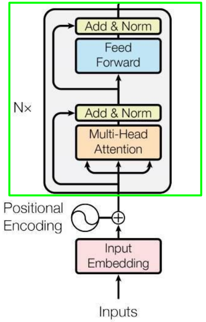
- Masked pre-training strategy -> downstream fine-tuning



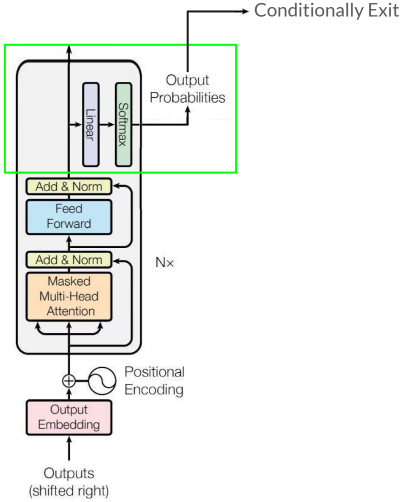
Architecture



Encoder-Decoder Transformer



BERT / Encoder



Classify at Each Layer

Training

Training Regime:

1. Identical pre-training as in BERT
2. Identical fine-tuning as in BERT
3. Freeze fine-tuned parameters, optimize intermediate off-ramps (classifiers)

Loss: cross-entropy loss for each off-ramp

$$L_i(\mathcal{D};\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} H(y, f_i(x;\theta)),$$

Inference

- Define Entropy threshold *S*

- Stop if offramp entropy < S
  - Entropy ~== uncertainty

Algorithm 1 DeeBERT Inference (Input: x)

for $i = 1$ to n **do**
  $z_i = f_i(x; \theta)$
  if entropy$(z_i) < S$ **then**
    return z_i
  end if
end for
return z_n

Experimental Setup

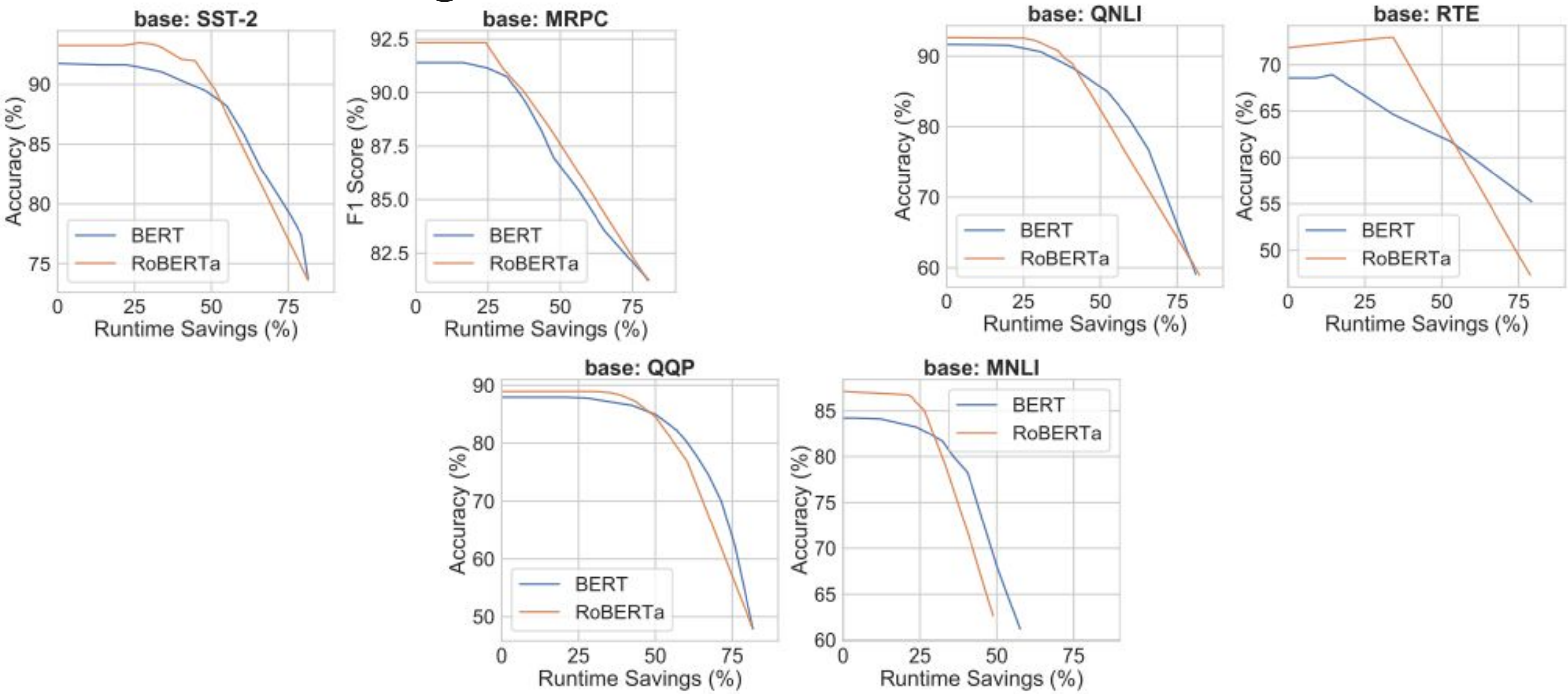
- DeeBert applied to pretrained BERT and RoBERTa models

- 6 classification datasets from GLUE benchmark
  - SST-2, MRPC, QNLI, RTE, QQP, and MNLI

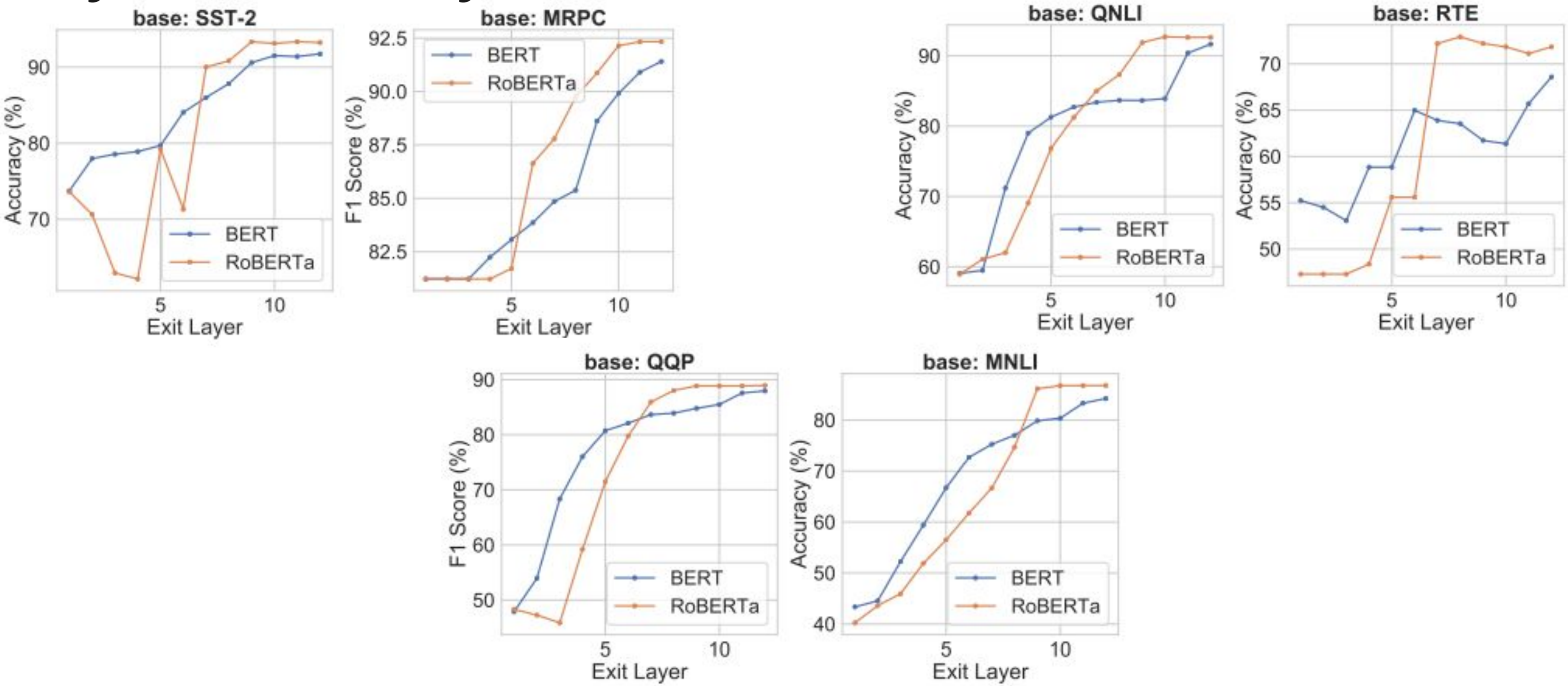
Experimental Results

	SST-2		MRPC		QNLI		RTE		QQP		MNLI-(m/mm)	
	Acc	Time	F_1	Time	Acc	Time	Acc	Time	F_1	Time	Acc	Time
BERT-base												
Baseline	93.6	36.72s	88.2	34.77s	91.0	111.44s	69.9	61.26s	71.4	145min	83.9/83.0	202.84s
DistilBERT	−1.4	−40%	−1.1	−40%	−2.6	−40%	−9.4	−40%	−1.1	−40%	−4.5	−40%
DeeBERT	−0.2	−21%	−0.3	−14%	−0.1	−15%	−0.4	−9%	−0.0	−24%	−0.0/−0.1	−14%
	−0.6	−40%	−1.3	−31%	−0.7	−29%	−0.6	−11%	−0.1	−39%	−0.8/−0.7	−25%
	−2.1	−47%	−3.0	−44%	−3.1	−44%	−3.2	−33%	−2.0	−49%	−3.9/−3.8	−37%
RoBERTa-base												
Baseline	94.3	36.73s	90.4	35.24s	92.4	112.96s	67.5	60.14s	71.8	152min	87.0/86.3	198.52s
LayerDrop	−1.8	−50%	-	-	-	-	-	-	-	-	−4.1	−50%
DeeBERT	+0.1	−26%	+0.1	−25%	−0.1	−25%	−0.6	−32%	+0.1	−32%	−0.0/−0.0	−19%
	−0.0	−33%	+0.2	−28%	−0.5	−30%	−0.4	−33%	−0.0	−39%	−0.1/−0.3	−23%
	−1.8	−44%	−1.1	−38%	−2.5	−39%	−1.1	−35%	−0.6	−44%	−3.9/−4.1	−29%

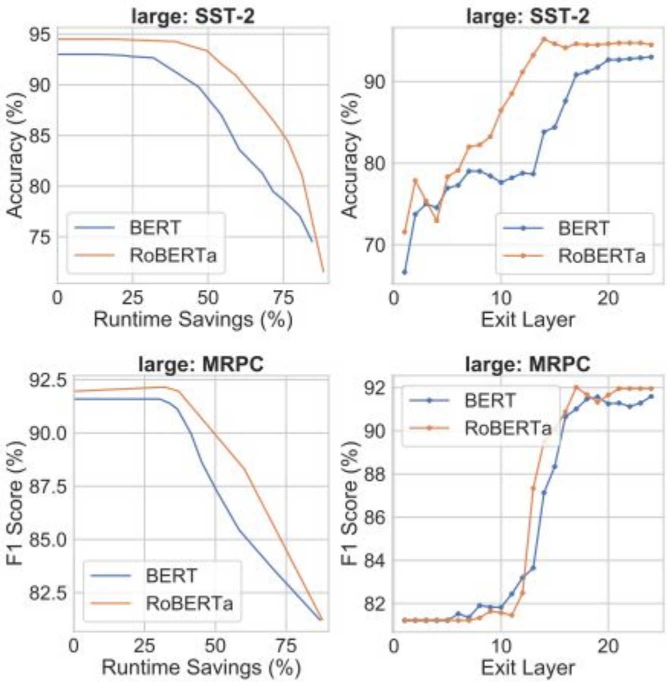
Runtime Savings



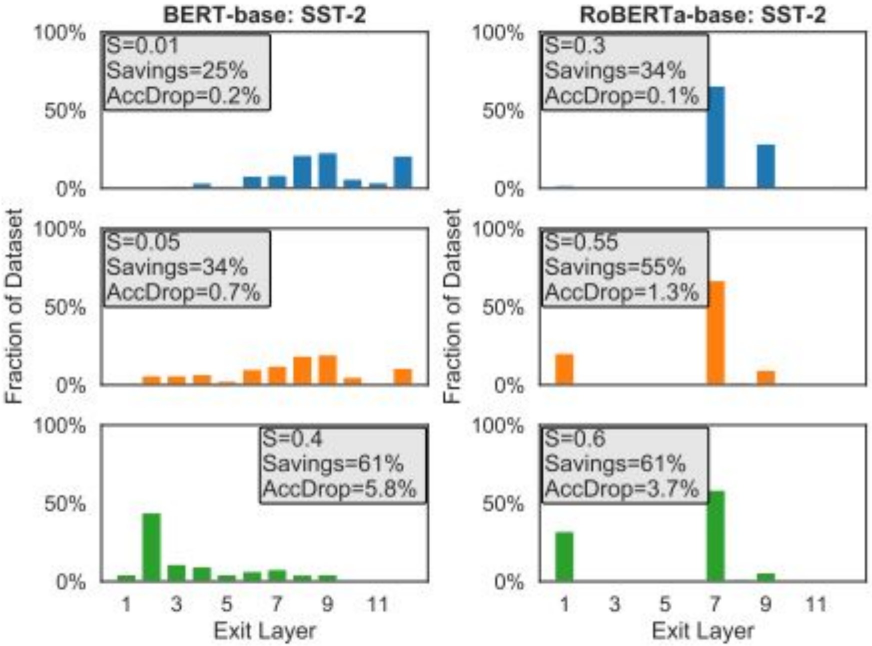
Layerwise Analysis



Impact on BERT-Large and RoBERTa-Large



Layer Exiting Proportion



Conclusion

- DeeBERT accelerates BERT & RoBERTa inference by up to ~40%
- Minimal performance loss
- Comparatively inexpensive additional training