Paper Reading

Xinwei Geng

2018-09-18

Breaking the Beam Search Curse: A Study of (Re-)Scoring Methods and Stopping Criteria for Neural Machine Translation

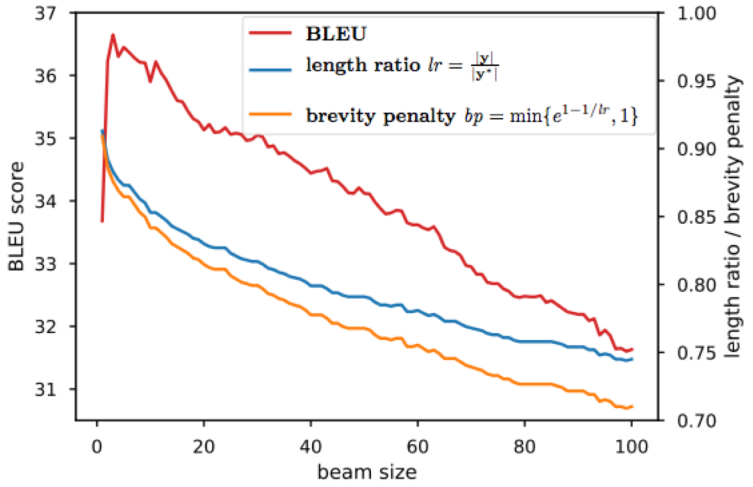
Yilin Yang, Liang Huang, Mingbo Ma

EMNLP 2018

Motivation

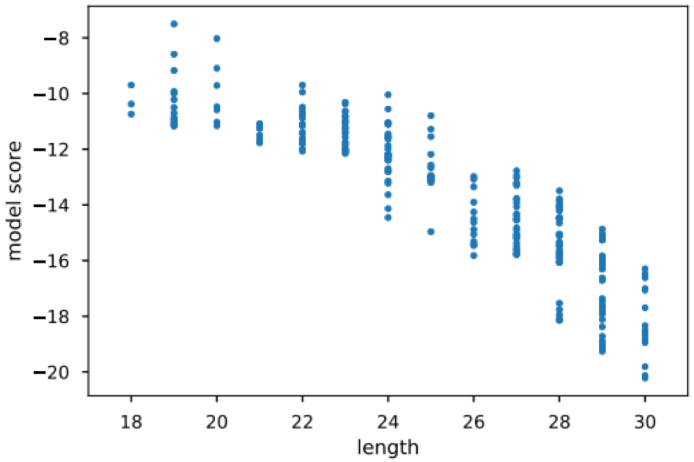
- Beam search is widely used in neural machine translation, and usually improves translation quality compared to greedy search.
- It has been widely observed however, beam sizes larger than 5 hurt translation quality
- With beam size increasing, |y| decreases, which causes the length ratio to drop

$$\text{BLEU} = bp \cdot \exp(1/4 \sum_{n=1}^{4} \log p_n)$$
$$\text{where} \quad bp = \min\{e^{1-1/lr}, 1\}$$
$$\text{where} \quad lr = |\mathbf{y}|/|\mathbf{y}^*|$$



Reasons

- As beam size increases, the more candidates it could explore
- Shorter candidates have clear advantages w.r.t. model score



Previous Rescoring Methods

- RNNSearch

$$\hat{S}_{\text{length\_norm}}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y})/|\mathbf{y}|$$

- Baidu NMT

$$\hat{S}_{\text{WR}}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y}) + r \cdot |\mathbf{y}|$$

- Bounded Word-Reward

$$\hat{S}_{\text{BWR}}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y}) + r \cdot L(\mathbf{x}, \mathbf{y})$$

$$L(\mathbf{x}, \mathbf{y}) = \min\{|\mathbf{y}|, gr \cdot |\mathbf{x}|\}:$$

Rescoring with length prediction

- we use a 2-layer MLP, which takes the mean of source hidden states as input, to predict the generation ratio
- Boundary word-Reward

$$L^*(\mathbf{x}, \mathbf{y}) = \min\{|\mathbf{y}|, L_{pred}(\mathbf{x})\}$$
$$\hat{S}_{\mathrm{BWR}^*}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y}) + r \cdot L^*(\mathbf{x}, \mathbf{y})$$

- Bounded Adaptive-Reward

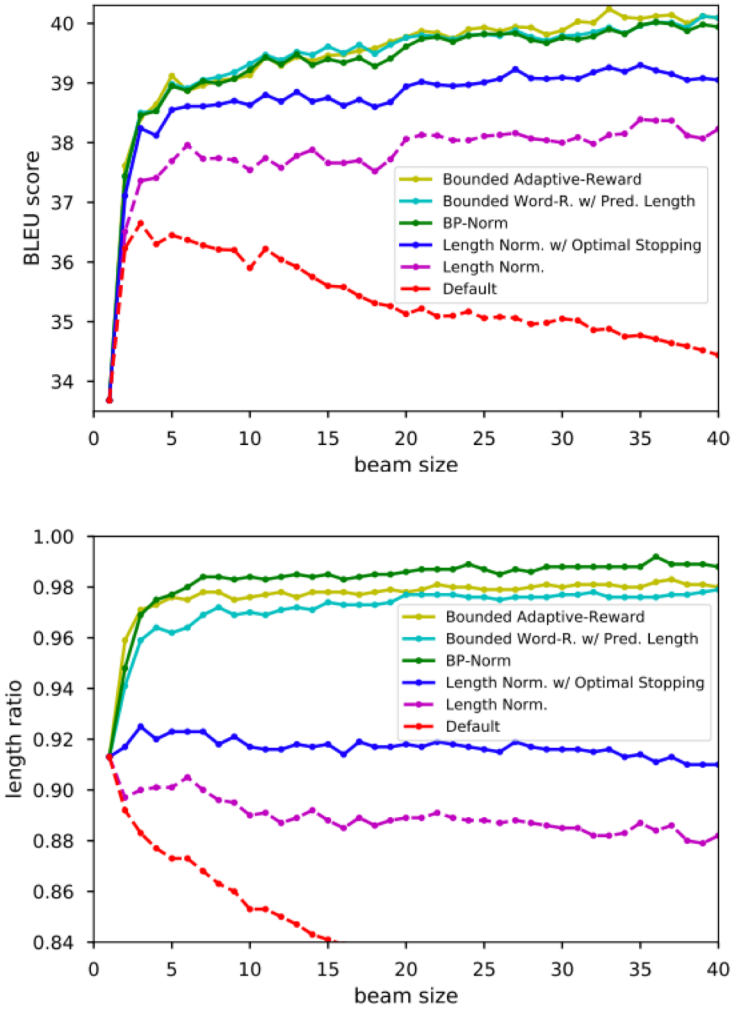
$$r_t = -(1/b) \sum_{i=1}^{b} \log p(\mathrm{word}_i)$$

$$\hat{S}_{\mathrm{AdaR}}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y}) + \sum_{t=1}^{L^*} r_t$$

- BP-Norm

$$\hat{S}_{bp}(\mathbf{x}, \mathbf{y}) = \log bp + S(\mathbf{x}, \mathbf{y})/|\mathbf{y}|$$

Length Ratios



Experiment Results

Small beam ($b = 14, 15, 16$)	dev		test	
	BLEU	ratio	BLEU	ratio
Moses (b=70)	30.14	-	29.41	-
Default (b=5)	36.45	0.87	32.88	0.87
Length Norm.	37.73	0.89	34.07	0.89
+ optimal stopping*	38.69	0.92	35.00	0.92
Wu et al. (2016) α=β=0.3	38.12	0.89	34.26	0.89
Bounded word-r. r=1.3	39.22	0.98	35.76	0.98
with predicted length				
Bounded word-r. r=1.4*	**39.53**	0.97	35.81	0.97
Bounded adaptive-reward*	39.44	0.98	35.75	0.98
BP-Norm*	39.35	0.98	**35.84**	0.99

Large beam ($b = 39, 40, 41$)	dev		test	
	BLEU	ratio	BLEU	ratio
Moses (b=70)	30.14	-	29.41	-
Default (b=5)	36.45	0.87	32.88	0.87
Length Norm.	38.15	0.88	34.26	0.88
+ optimal stopping*	39.07	0.91	35.14	0.91
Wu et al. (2016) α=β=0.3	38.40	0.89	34.41	0.88
Bounded word-r. r=1.3	39.60	0.98	35.98	0.98
with predicted length				
Bounded word-r. r=1.4*	40.11	0.98	36.13	0.97
Bounded adaptive-reward*	**40.14**	0.98	**36.23**	0.98
BP-Norm*	39.97	0.99	**36.22**	0.99

Table 1: Average BLEU scores and length ratios over small and large beams. $\star$ indicates our methods.

Speeding Up Neural Machine Translation Decoding by Cube Pruning

Wen Zhang, Liang Huang, Yang Feng, Lei Shen, Qun Liu
EMNLP 2018

Drawbacks

- Although neural machine translation has achieved promising results, it suffers from slow translation speed.

	GPU		CPU	
Calculation Units	**Time(s)**	**Percentage**	**Time(s)**	**Percentage**
Eq. (6): $s_j = f(e_{y^*_{j-1}}, s_{j-1}, c_j)$	551.07	75.73%	1370.92	19.42%
Eq. (7): $t_j = g(e_{y^*_{j-1}}, c_j, s_j)$	88.25	12.13%	277.76	3.93%
Eq. (8): $o_j = \mathbf{W}_o t_j$	25.33	3.48%	2342.53	33.18%
Eq. (9): $\mathcal{D}_j = \mathrm{softmax}(o_j)$	63.00	8.66%	3069.25	43.47%

Self-normalization

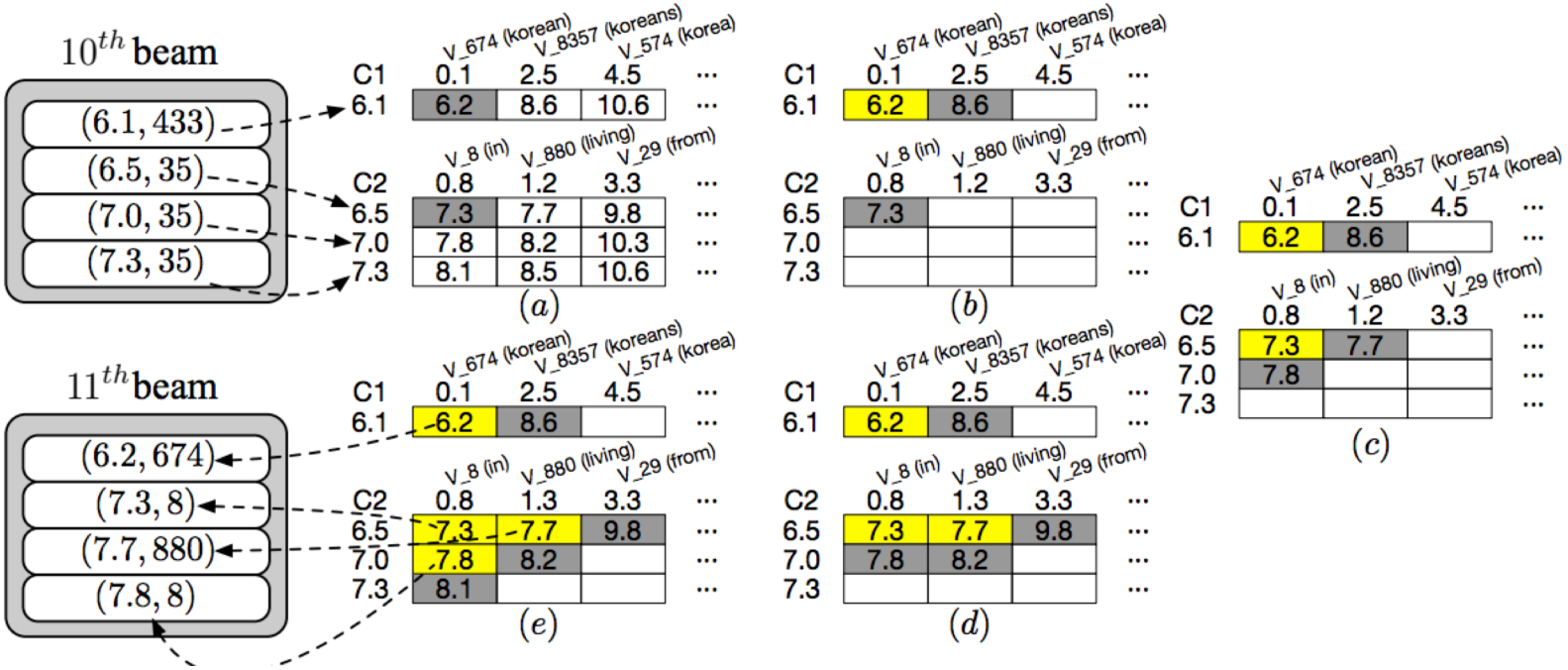
- approximate the probability distribution over the target vocabulary without normalization operation

$$
\begin{aligned}
\mathrm{L}_\theta &= -\sum_{j=1}^{|\mathbf{y}|} \log \mathcal{D}_j[y_j^*] \\
&= -\sum_{j=1}^{|\mathbf{y}|} \log \frac{\exp\left(o_j[y_j^*]\right)}{\sum_{y' \in V} \exp\left(o_j[y']\right)} \\
&= \sum_{j=1}^{|\mathbf{y}|} \log \sum_{y' \in V} \exp\left(o_j[y']\right) - o_j[y_j^*]
\end{aligned}
\tag{10}
$$

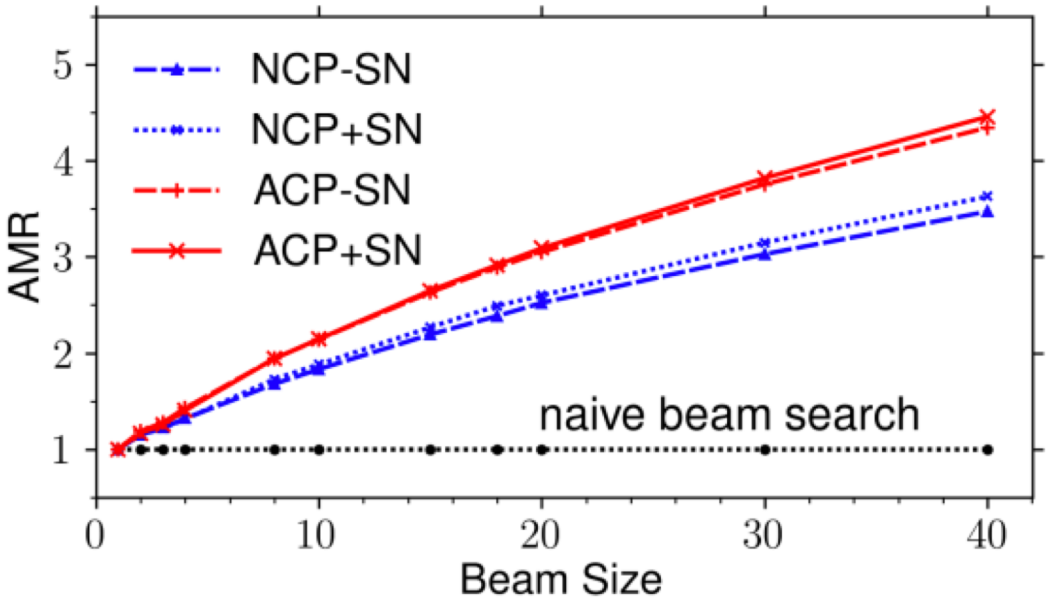
$$
\begin{aligned}
\mathrm{L}_\theta &= -\sum_{j=1}^{|\mathbf{y}|} \left(\log \mathcal{D}_j[y_j^*] - \alpha(\log Z - 0)^2\right) \\
&= -\sum_{j=1}^{|\mathbf{y}|} \left(\log \mathcal{D}_j[y_j^*] - \alpha \log^2 Z\right)
\end{aligned}
\tag{1}
$$

Cube Pruning

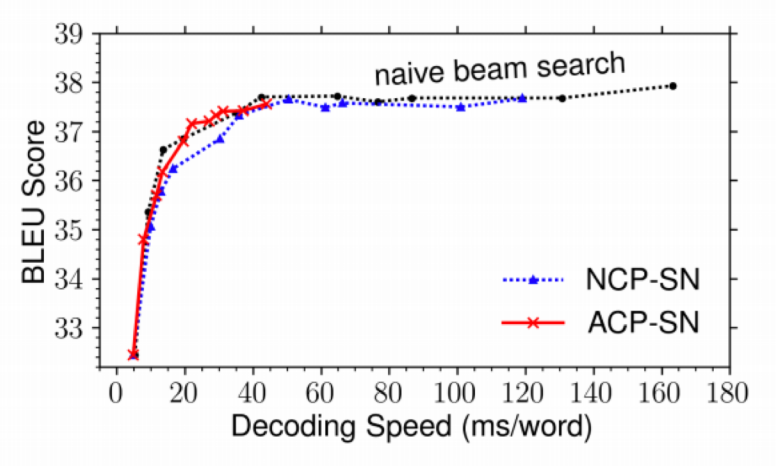
- items being merged in the previous beam should have the same target word



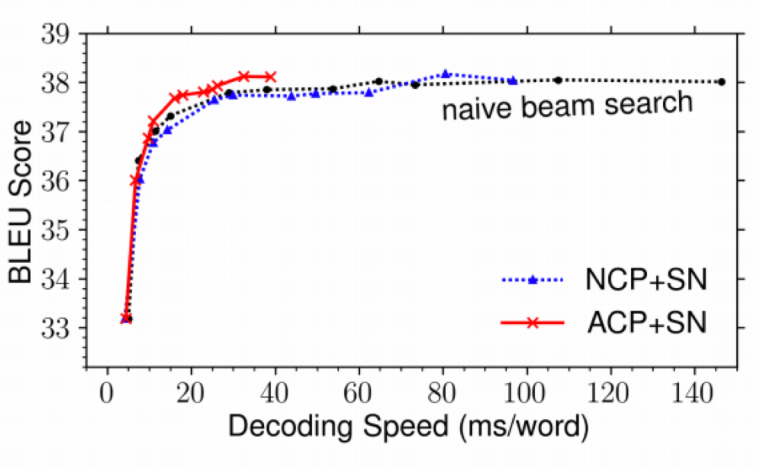
Average Merge Rate



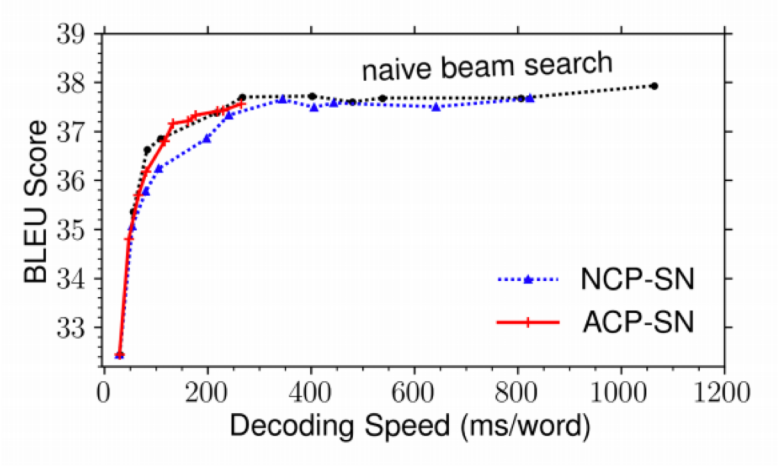
Decoding speed

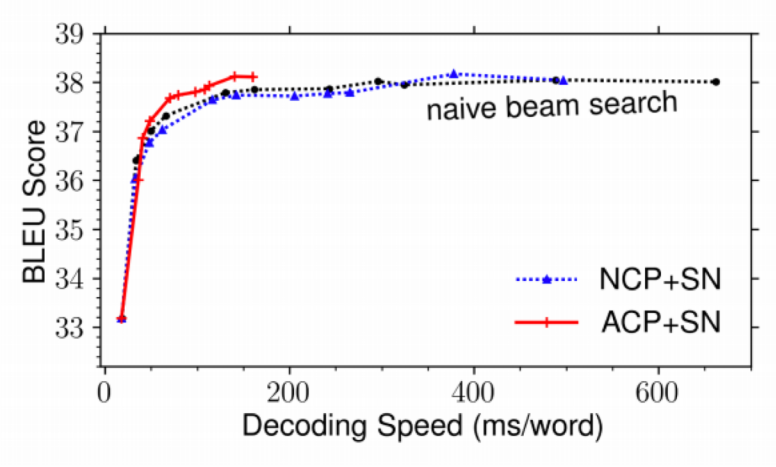


(a) BLEU vs. decoding speed, without self-normalization

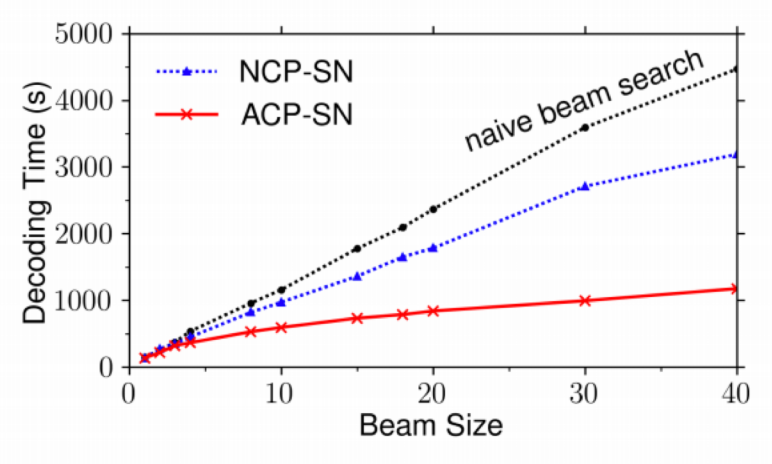


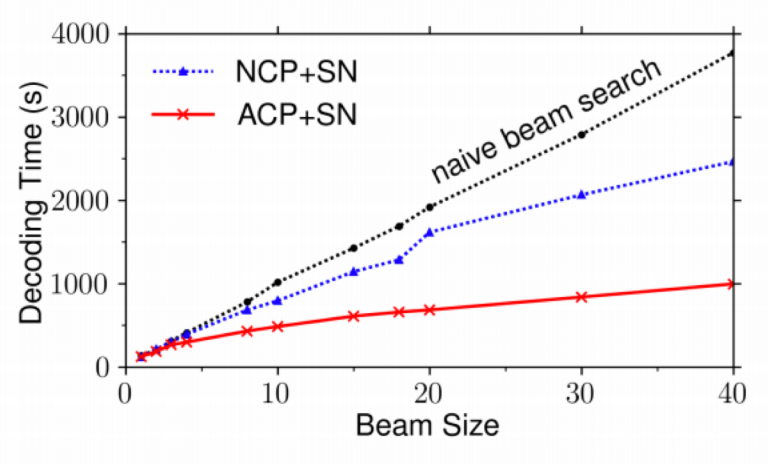
(b) BLEU vs. decoding speed, with self-normalization

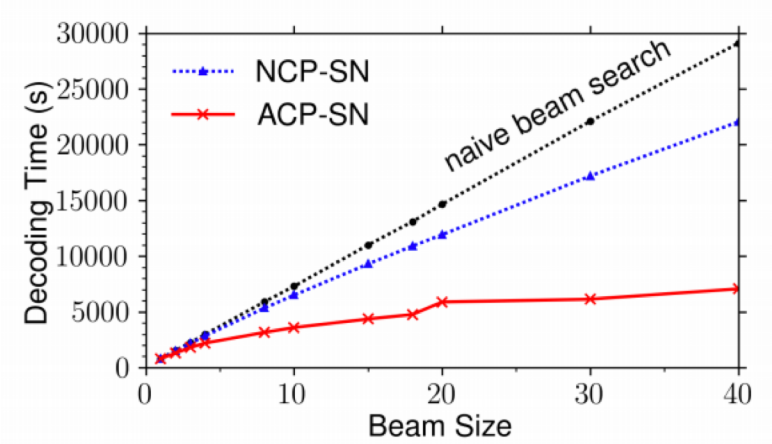


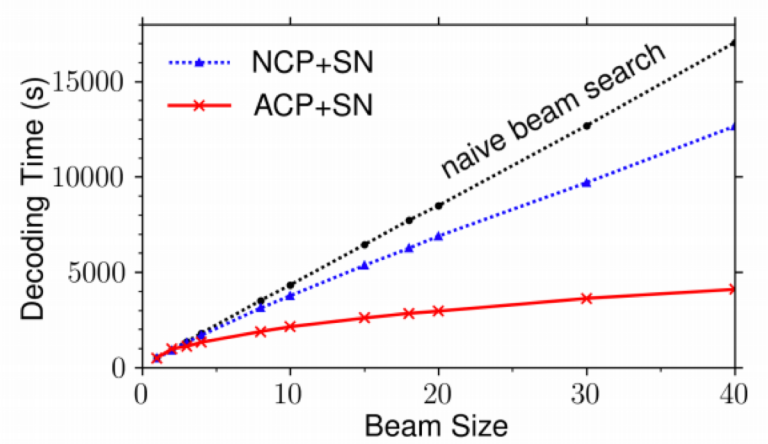


Time consumption









Thanks & QA