A Simple and Effective Approach to Coverage-Aware Neural Machine Translation

Yanyang Li, Tong Xiao, Yinqiao Li, Qiang Wang, Changming Xu
Natural Language Processing Lab., Northeastern University

Xueqiang Lu
Beijing Key Laboratory of Internet Culture and Digital Dissemination Research

Motivation

- Standard NMT model score prefers shorter translations, because the log-probability is added over time steps:

$$\log P(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{|\mathbf{y}|} \log P(\mathrm{y}_j|y_{<j}, \mathbf{x})$$

- Normalizing the model scores by lengths eliminates this system bias

But it is coverage-unaware method and thus not able to distinguish content-rich translations.

Coverage: the extent of a source word is translated

Source: 你 知道 去 北京站 的 路 怎么 走 吗

Target-1: Do you know the way <EOS>

-1.5 + -2.5 + -1.3 + -2.1 + -1.5 + -2.0 = -10.9 (length=6)

Same length, but target-3 is better :(

Target-2: to go to Station <EOS>

-2.1 + -1.7 + -1.6 + -1.4 + -1.9 + -2.0 = -17.9

Target-3: Beijing Railway Station <EOS>

+ -1..3 + -1.9 + -2.1 = -18.0 (length=10)

	Score	Score with LN
Target-1	-10.9	-10.9 / 6 = -1.82
Target-2	-17.9	-17.9 / 10 = -1.79 √
Target-3	-18.0	-18.0 / 10 = -1.80

Note: d=0, w=1.0

Approach

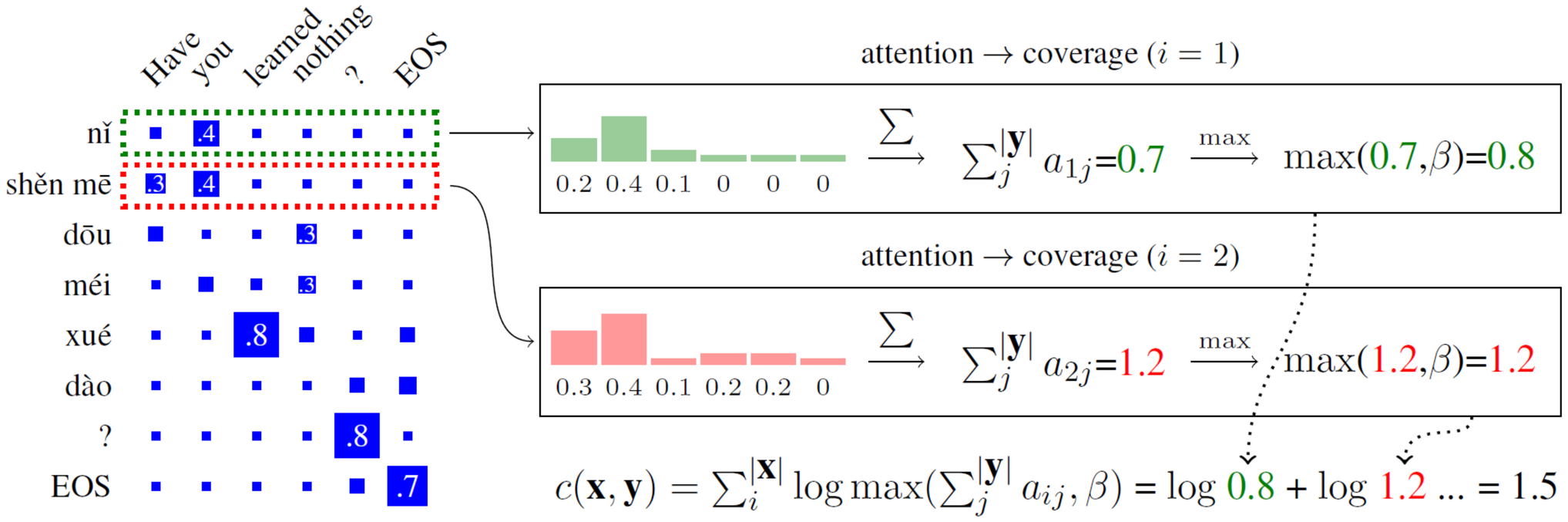
The basic idea is to apply coverage-sensitive feature (Coverage Score) at every decoding step

- Coverage Score in Beam Search
  1. Compute coverage score from attention a
     $c(\mathbf{x}, \mathbf{y}) = \sum_i^{|\mathbf{x}|} \log \max(\sum_j^{|\mathbf{y}|} a_{ij}, \beta)$
  2. Combined with the model score
     $s(\mathbf{x}, \mathbf{y}) = (1-\alpha) \cdot \log P(\mathbf{y}|\mathbf{x}) + \alpha \cdot c(\mathbf{x}, \mathbf{y})$
  3. Applied in Beam Search

- Comparison of different coverage-sensitive methods

Method	Model-Free	Beam Search
Coverage Model	×	√
Coverage Penalty	√	×
*Coverage Score	√	√

- Coverage Score for a running example (Chinese pinyin-English and $\beta = 0.8$)



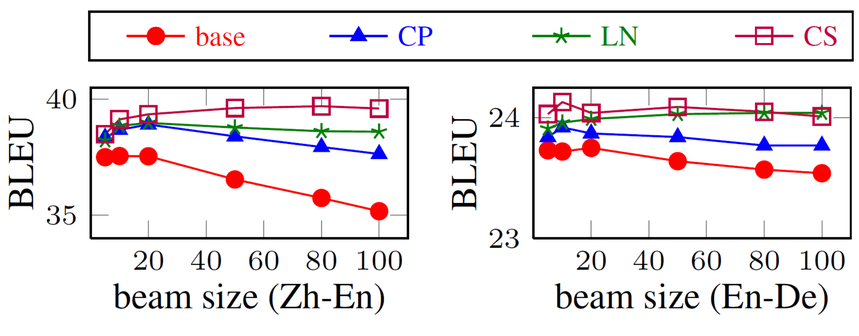
Experiments

Entry	beam=10				beam=100				beam=500			
	Zh-En		En-De		Zh-En		En-De		Zh-En		En-De	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
base	37.55	30.91	23.72	23.36	35.17	28.48	23.54	23.50	23.40	17.95	23.15	23.24
LN	38.85	32.32	23.96	22.96	38.60	31.97	24.04	23.14	37.60	30.81	23.95	23.16
CP	38.68	31.32	23.96	23.27	37.64	30.82	23.77	23.65	34.81	28.82	23.43	23.46
CP†	35.96	29.98	23.67	23.53	34.77	27.45	23.69	23.63	32.23	25.09	23.65	23.61
LN+CP	39.07	32.47	23.98	23.26	38.93	32.39	23.95	23.60	37.88	31.46	23.77	23.64
CS	39.13	32.24	24.13	**23.62**	39.60	32.71	24.01	**23.84**	39.50	32.77	23.96	**23.85**
CS†	38.76	32.18	24.18	23.30	37.79	31.57	23.99	23.75	35.89	29.92	23.75	23.70
LN+CS	39.59	32.73	24.24	23.32	39.88	33.20	24.22	23.60	**39.77**	**32.89**	**24.17**	23.57
LN+CP+CS	**39.62**	**32.75**	**24.27**	23.30	**39.90**	**33.23**	**24.24**	23.65	39.73	32.85	24.17	23.69

CP†: Coverage Penalty at each decoding step; CS†: Coverage Score only in reranking

	dev	test
Zh-En	MT06	MT08
En-De	news13	news14

- Coverage Score performs better than other approaches
- Coverage Score performs the best with other approaches
- Coverage Score performs the best inside Beam Search
- Coverage Score performs better with larger beam sizes



← Coverage Score is robust to beam size

→ Coverage Score is robust to sentence length