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

Yanyang Li[1], **Tong Xiao**[1], **Yinqiao Li**[1], **Qiang Wang**[1], **Changming Xu**[1] and **Xueqiang Lu**[2]
[1]Natural Language Processing Lab., Northeastern University
[2]Beijing Key Laboratory of Internet Culture and Digital Dissemination Research
blamedrlee@outlook.com, xiaotong@mail.neu.edu.cn, li.yin.qiao.2012@hotmail.com, wangqiangneu@gmail.com, changmingxu@neuq.edu.cn, lxq@bistu.edu.cn

A Decoding Hyperparameters

Here we describe the details of strategy to combine and tune *length normalization* (LN), *coverage penalty* (CP) and our *coverage score* (CS), and the corresponding hyperparameters settings that we used in experiments.

A.1 Combination Strategies

To combine LN and CS, we use Eq. (1) for each time step. The first term of Eq. (1) denotes the standard log-likelihood normalized by LN. The second term is CS divided by the length of source sentence $|\mathbf{x}|$. This division is a form of normalization to preserve similar scale as the normalized log-likelihood because the normalized log-likelihood might no longer decline as decoding proceeded, while the raw coverage score would increase and lower the performances. Since CS is the sum of log scores over $\mathbf{x}$-axis, it is divided by the length of source sentence $|\mathbf{x}|$ instead of target sentence $|\mathbf{y}|$. Finally we linearly interpolate these two scores together for hypotheses comparison during beam search.

$$s(\mathbf{x},\mathbf{y}) = (1-\alpha)\frac{\log \mathrm{P}(\mathbf{y}|\mathbf{x})}{\mathrm{LN}(|\mathbf{y}|,w)} + \alpha\frac{c(\mathbf{x},\mathbf{y})}{|\mathbf{x}|} \quad (1)$$

Similar to LN, exponentially rescaling $|\mathbf{x}|$ by a separate tunable hyperparameters is possible, and in our preliminary experiments, dividing CS by LN with the shared w (See Eq. (2)) has slightly better performances (+0.2 BLEU) in Zh-En translation, which might benefits from the tunable LN weight w, but the simple form of division in Eq. (1) works sufficiently well in both translation tasks with the less tuning burden.

$$s(\mathbf{x},\mathbf{y}) = \frac{(1-\alpha)\cdot\log \mathrm{P}(\mathbf{y}|\mathbf{x}) + \alpha\cdot c(\mathbf{x},\mathbf{y})}{\mathrm{LN}(|\mathbf{y}|,w)} \quad (2)$$

To combine LN, CP and CS, we use Eq. (1) in beam search, and use Eq. (3) for reranking. Because CP is effective in finished hypotheses space, it only appears in reranking, i.e., Eq. (3), thus the score used in each beam search step is exactly the one we used to combine LN and CS, i.e., Eq (1).

$$s(\mathbf{x},\mathbf{y}) = \frac{\log \mathrm{P}(\mathbf{y}|\mathbf{x})}{\mathrm{LN}(|\mathbf{y}|,w)} + \frac{\alpha}{1-\alpha}\frac{c(\mathbf{x},\mathbf{y})}{|\mathbf{x}|} + \lambda\frac{\mathrm{CP}(\mathbf{x},\mathbf{y})}{|\mathbf{x}|} \quad (3)$$

The first term of Eq. (3) denotes the standard log-likelihood normalized by LN. The second term is normalized CS as in Eq. (1). The third term is CP divided by the length of source sentence $|\mathbf{x}|$. Similar to CS, we normalize CP to boost the performance. For the second term, its coefficient is $\frac{\alpha}{1-\alpha}$ rather than α, and the coefficient of the first term is constant 1. We expect that moving the linear interpolation weight $1-\alpha$ from the log-probability to CS can remove the dependence between α and λ and thus ease hyperparameters tuning, while it still preserves the same proportion of normalized log-likelihood to normalized CS for both decoding and reranking.

A.2 Experiments Settings

Table 1 is the hyperparameters settings of both Chinese-English and English-German translation tasks for different combinations of decoding heuristics. w is the hyperparameter of LN, λ is the weight of CP, α is the linear interpolation weight of CS and β is the truncation threshold of CS.

A.3 Tuning Details

For hyperparameters tuning for the combination of all approaches, the special construction of the combination strategy, as shown in both Eq. (1) and Eq. (3), introduces some sort of independence among these features, e.g., it enforces the scale

	Entry	Hyperparameters			
		w	λ	α	β
Zh-En	CP†	-	0.1	-	-
	CS†	-	-	0.6	0.3
	LN+CP	1.4	0.1	-	-
	LN+CS	1.2	-	0.7	0.4
	LN+CP+CS	1.2	0.4	0.7	0.4
En-De	CP†	-	0.1	-	-
	CS†	-	-	0.3	0.2
	LN+CP	0.4	0.1	-	-
	LN+CS	1.2	-	0.5	0.1
	LN+CP+CS	1.0	0.8	0.6	0.2

Table 1: Decoding Hyperparameters Settings.

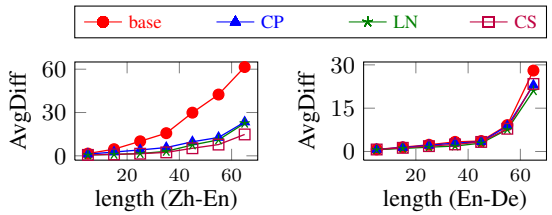

Figure 1: AvgDiff against sentence length.

between LN and normalized CS to be similar to the raw log-likelihood and CS by dividing sentence length. This fact alleviates the tuning burden through reusing or fine-tuning hyperparameters determined by simpler experiments in a small range, e.g., LN+CS experiment can reuse α and β setting from CS experiment, which is also convinced by Table 1.

A subtle detail of grid search is that if we found the optimal value appears in the boundary of the interval, e.g., $w = 0.5$ for $[0.5, 1.5]$ in LN+CP experiment, we slightly extend the interval for evaluation of tuning to seek a setting with better performance, e.g., $[0.5, 1.5] \rightarrow [0.3, 1.5]$ in LN+CP experiment.

Besides, some interesting observations from our experiments might help to prune out unnecessary searching branches, e.g., β of CS usually works well with small values, so we can search in a small interval like $[0, 0.5]$, and CP (without normalization) works only with small weights, so we can evaluate it with weights within $[0, 0.5]$.

B Translation Length

Figure 1 compares the average translation lengths for different source sentence lengths. For longer source sentences, there is a larger margin of length difference between translations and references,

	Entry	Zh-En			
		MT06	MT04	MT05	MT08
b=10	base	37.55	42.60	35.96	30.91
	LN	38.85	44.31	37.85	**32.32**
	CP	38.68	44.10	37.73	31.84
	CS	**39.13**	**45.03**	**38.39**	32.24
b=100	base	35.17	39.61	33.19	28.48
	LN	38.60	43.73	37.79	31.97
	CP	37.64	42.70	36.45	30.82
	CS	**39.60**	**45.84**	**39.54**	**32.71**
b=500	base	23.40	25.49	20.60	17.95
	LN	37.60	40.25	36.60	30.81
	CP	34.81	38.01	33.74	28.82
	base‡	37.26	43.51	37.57	31.24
	LN‡	38.60	43.43	37.60	31.20
	CP‡	39.38	44.42	38.00	31.76
	CS	**39.50**	**45.48**	**39.78**	**32.77**
	CS†	35.89	40.14	35.24	29.92

Table 2: BLEU against different datasets and different beam sizes. ‡ denotes that it is searched by beam search with coverage score (`CS`), then reranked by log-probabilities (`base`), length normalization (`LN`) and coverage penalty (`CP`) respectively. † denotes that it is searched by standard beam search, then reranked by `CS`.

which implies that under-translation is more likely to appear in long sentences. It shows that the reason why NMT is unable to process long sentences well is mainly due to the difficulty of generating translations with proper lengths for long sentences. As shown in Figure 1, LN and CP can alleviate this problem to some extent, but our approach can do this better.

C More Results

Table 2 shows more experiment results, and our method still outperforms other approaches in almost all datasets and beam sizes. The ‡ lines in Table 2 show that ranking hypotheses with n-best list generated by our method CS can improve other methods performances, which implies that our method generates better n-best list no matter which metric is used to measure. The † line shows that our proposed method can rank n-best list well, and can receive more gain if it is applied to each decoding step.